2023

Latis Design Review Report

Faculty Mentor: Dr. Natalie Cherbaka & Matt Earnest

SOHAM GANDHI, ISABELLA JENSEN, FORREST MENG, DIYA WADHERA, SOMYA DUBEY, PRIYA CHINNAREDDYVARI, & RAKESH PILLAI, CAMILLE D'AMICO

Table of Contents

Executive Summary

Currently, the Internet of Things (IoT) encompasses the connection between approximately 27 billion connected devices. Devices can be integrated into the IoT relatively easily, with the use of affordable computer chips and ubiquitous networks. The contents of this report will be considering the cybersecurity impacts of this move into the IoT, including how device security, or the lack thereof, is impactful to one specific area, the production line.

Researching, and thoroughly addressing, the complete device security problem space presented a daunting and unattainable task. Amid this process, research was guided through conversations with stakeholders and researchers at Virginia Tech. From here, the scope of LATIS' project was formed, and the question soon evolved: how to facilitate IoT device security in the manufacturing sector?

However, at this point, there was still much to refine concerning scope. Primary data collection at this stage occurred through interviews with a diverse set of stakeholders in the industrial cybersecurity sector. Interviews led LATIS to the consideration of numerous problem spaces, including securing manufacturing geometries and, the current problem space, streamlining firmware updates for large actuators on production lines, which received considerable stakeholder support.

Using a four sets approach, the remainder of this report defines and conducts an in-depth analysis of the articulated problem space. The coming introductory pages provide a deeply researched summary of specific technologies in the problem space, their interaction using the IoT, how these are vulnerable to cyber-attacks, and existing solutions. Next, the report a potential solution we plan on utilizing and evaluating the criteria's to ensure we meet stakeholder needs. The final section of this report discusses the implementation details of this idea and how we plan on testing this system in real-world scenarios.

In our solution concept, distributed ledger technology (DLT) is utilized to verify the identity of a user sending an update and allows devices on the IoT to verify that the update is valid. The systems involved in this process are thoroughly explained in the solution concept segment of this document, where a visual representation of the solution concept and involved enabling technologies is presented as well as a thorough run down of the backend of the system. This section contains an example of a firmware update being transferred over the air, detailing the process involvement of DLT, encryption, and smart contracts.[1]

[1] Executive summary - Diya, Soham

Introduction

Problem Space

Operational Technologies

Manufacturing and operational facilities, such as Boeing's factories and natural resource treatment plants, all use operational technologies to help with operating, monitoring, maintaining, and updating technology that interacts with the physical environment throughout the facilities. Devices that fall under the jurisdiction of OT include robotic arms, CNC machines, treadmills, small sensors, and many more.

OT includes an ICS (Industrial Control System) pyramid hierarchy. At the bottom layer, there are I/O devices, such as actuators, sensors, and end effectors, where actual data is sent and interpreted on the edge to perform and monitor tasks. The next level up is PLCs (Programmable Logic Controllers) and RTUs (Remote Terminal Units). These devices interface with the actual devices. PLCs take data from sensors via input channels and control actuators via output channels and are mostly local to the device or machine. Meanwhile, RTUs allows for remote control and monitoring of a device or system by taking data from the machine, sending it as telemetry to a master controller, and taking input commands from the controller as well. Both are microcontrollers with an array of I/O channels and can be add-on cards that attach to the backplane of the computer of the machine. Programming of these devices can be done through USB, Serial Interfaces, or via the network.

This master controller is the next step in the hierarchy. SCADA (Supervisory Control and Data Acquisition) and DCS (Distributed Control System) are two frameworks that help with the control and data acquisition of systems, machines, and devices in or across facilities. SCADA describes a system of a combined use of ICS and edge devices working on a common task, and currently, there are software systems that help human operators visualize the entire system. This allows for a better scope of control for the operational tasks. Other layers could be added on for more robust monitoring and automated control, such as PCS (Process Control Systems), EMS (Energy Management Systems), and SIS (Safety Instrumented Systems). Above the SCADA level of data acquisition are the Production Control and Enterprise levels, which oversee all facility operations and provide high-level decision support. Our scope for this problem space is mainly between the PLC/RTU level and the SCADA level.

The increased interconnectivity has led to a rise in "Industry 4.0," where various processes of a facility are connected through a network. This has led to the growth of IoT (Internet of Things), which we are defining as any device that is connected to some sort of network, whether it be the internet, an intranet, or a cloud platform. Our problem space definition also includes large-scale actuator machines as the IoT device, as they can cause more disruption to a supply chain due to failure and usually have PLC/RTUs installed.[2]

[2] Operational Technologies - Forrest

Communication

With the rise of "Industry 4.0," the importance of communication protocols has increased significantly since many companies default to these protocols to secure communications as found from interviews with industry experts at MOOG and DoD contracting companies.

In the industry, there are three types of protocols: communication, management, and security. Communication protocols, such as TCP/IP and HTTP, are the most basic and commonly used for general communication and web browsing. Management protocols maintain and govern the network, while security protocols offer additional security features to prevent cyber-attacks (3 Main Types of Network Protocols, Explained, 2022).

TCP/IP and HTTP are included in the communication protocol group, where TCP/IP governs network communication and is the most common protocol used in this category. These protocols are widely used, but they are also the most vulnerable to cyber-attacks (Internet, TCP/IP, and HTTP Concepts, 2020).

Management protocols, such as ICMP and SNMP, are used for debugging issues in a network and measuring network performance. ICMP detects DDoS attacks and determines network latency, while SNMP measures device resource usage and detects potential cyber-attacks by spotting anomalies in resource usage (What Is ICMP? | Internet Control Message Protocol, n.d.; SNMP Ports & Protocol - What Is It?, n.d.).

Security protocols, including HTTPs, SFTP, and SSL, are designed to prevent unwanted traffic and data breaches. HTTPs is used by modern websites and is similar to HTTP, but with an SSL certificate that enhances security features. SFTP, on the other hand, is a file transfer protocol secured with software encryption algorithms like AES (The Difference Between HTTPS and SFTP, 2022). While these protocols are best for mainstream systems, they still have vulnerabilities to attacks.[3]

Cyber Attacks

Increased communication and devices in a system lead to a larger "attack surface" or amount of entry points for a bad actor. Due to VPNs and Firewalls, most bad actors are not able to penetrate the network. However, with phishing and internal bad actors, OT plants still need to be wary of attacks that can affect overall operations. Some of these relevant attacks are:

Rootkits: This malware type is designed to enable access to a part of the computer, such as the kernel, or an area of software that is otherwise not allowed while masking its existence. This allows an attacker to have a backdoor for other types of attacks. This also can allow the attacker to run other applications masked as the user, intercept installation vectors and change API behaviors, inject .dll scripts into target processes, run with the highest OS privileges, and hide processes by changing the kernel data structures of the OS. Detection of rootkits is practically impossible once the OS has been subverted; otherwise,

[3] Communication - Soham

behavior, signature, and difference-based detection are possible. To prevent rootkits, system hardening such as patches, secure boot, and the principle of least privilege/zero trust is recommended.

Man-in-the-middle: This attack is when the attacker sits between a transceiver and a receiver and relays communications between the two parties who think they are just communicating with each other. Active eavesdropping is the simplest version of a Man-in-the-middle attack, especially for wireless systems, as the attacker just needs to be in range to track the signal bands sent between parties. The middleman could also possibly alter the message, hiding the fact they had changed the contents of the message. According to an interview with Stephanie Travis from the Hume Center, man-in-the-middle attacks are a prevalent concern for defense as uncovering and modifying IP/secret information are active threats. Defense against man-in-the-middle attacks primarily relies on authentication, such as using a third-party certificate authority (CA) or HTTP Public Key Pinning.

Botnets: This attack is particularly risky for interconnected IoT devices. Using a zombie computer (one that is already infected with a rootkit or something similar), the botnet can spread through peer-to-peer communication either across the network or to a specific location. The controller of the botnet can direct the activities of the other computers through standard communication channels. Peer-to-peer botnets can perform as a command distribution server and a client that receives commands, avoiding a single point of failure. IRC and SMTP are two protocols that botnets favor.

Denial of Service: These attacks happen when a certain device, network, or program has its ability to run jammed or removed by a DoS program. Specifically for IoT, one of the most typical attacks is jamming in the wireless network to target specific packets in each layer. In a DoS/DDoS attack, the first step is scanning for vulnerabilities, then the attacker recruits machines to generate streams of packets to send to a vulnerable machine. The DDoS attack could send packets to different destinations with the target's IP address as the source address (called Reflection) or can send a large number of packets to one victim machine (Amplification). Both of these attacks tend to attack weaknesses in the TCP/IP communication layer. In other network layers, such as the 6LoWPAN layer (used often in Industrial IoT), DDoS could use fragment duplication as well as buffer reservation. To defend against DDoS attacks, rate limits, active filtering, IP traceback, ML algorithms, and decentralized ledger technologies are possible solutions.[4]

Recent Hacks

In the past couple of decades, cyber-attacks against manufacturing and SCADA systems have risen as IoT and internet connectivity also have exponentially increased. The staple example of an OT system being hacked and successfully attacked was the Stuxnet worm attack in 2010. Stuxnet is a self-replicable worm that started spreading through a USB stick plugged into a PLC in an Iranian nuclear power plant. This worm created a rootkit of the PLC's kernel and could spread between devices, using a stolen key from prominent embedded system Taiwanese manufacturers. This worm took advantage of several zero-day bugs in Windows OS-based machines, which allowed for privilege escalation and were not patched in the facility at the time, and eventually took out one-fifth of the plant's nuclear centrifuges (Fruhlinger, 2022).

[4] Cyber Attacks - Forrest

Several other major attacks happened in the years following. Night Dragon was a Trojan, which set many of the practices used in similar cyberattacks in the 2010s, that infected various HMIs and computers through a .dll file, exploiting Windows OS vulnerabilities to harvest proprietary information, dump data, and crack more hashes inside an entire SCADA system of several global oil, energy, and petrochemical companies (*McAfee Night Dragon Report (Update A) | CISA*, 2011). BlackEnergy was a DDoS malware that targeted SCADA HMI stations; the third iteration from a Russian cyberattack took out a major portion of the Ukrainian power grid in December of 2015 (*BlackEnergy APT Attacks | What Is BlackEnergy?*, n.d.). CRASHOVERRIDE was another malware targeted on ICS systems, specifically using the protocols IEC101, IEC104, and IEC61850, which are used in electric power control systems (*CrashOverride Malware | CISA*, 2017). It attacks the ICS legitimate control system, such as injecting commands into RTUs to weaken the overall grid reliability, denying COM access on Windows-based computers, exploiting a (now patched) Siemens delay DoS vulnerability, and can wipe OS's to become inert. Over-the-air updates have also been shown to be possibly vulnerable, as an Android systems manufacturer in China exposed over 3 million Android phones to Man-in-the-Middle attacks in 2016 (Arghire et al., 2016). These attacks against OT systems have not stopped even with security developments, as recently, Microsoft released a report on a vulnerability in one of their SDKs against supply chain IoT that allows for control over OT systems, one of which had targeted a water plant in India (*Vulnerable SDK Components Lead to Supply Chain Risks in IoT and OT Environments*, 2022). Even at Virginia Tech, according to an interview with Randy Marchany, CISO at Virginia Tech, there are thousands of probes every day to Virginia Tech's devices from international actors, seeing if any has a vulnerability to exploit.

Many of these attacks happened due to three factors: first, some sort of entryway through phishing or unauthorized access to a communication input outlet; second, an unsecured communication pathway; third, a vulnerability due to outdated OS or firmware systems. It is hard to secure the human weak link; therefore, our problem space lies within how to address the vulnerabilities of the actual microcomputer systems. [5]

What we want to solve

Our research shows that vulnerabilities in old firmware and operating systems in IoT devices are a primary enabling factor for cyberattacks against OT manufacturers' large-scale actuators. Although firewalls, network security practices, and file scanning for phishing attacks are good solutions to prevent bad actors from entering an OT environment, solutions for cybersecurity on the edge are relatively less developed and robust. By researching ways to ensure that manufacturers' systems are up to date and can be properly patched by trusted entities, we can help resolve many vulnerability issues that otherwise are sitting as a gateway for attackers to exploit. We focus on securing large-scale actuator IoT as they are usually devices with PLCs, RTUs, and computers, which could serve as zombie computers for attacking botnets. They also are the ones that, if compromised, can induce monetary loss and safety risks to both the operations and humans. [6]

[5] Recent Attacks - Forrest
[6] Problem Space - Forrest

Current Solutions

One company currently working on a monitoring solution for IoT devices is Zerynth. Zerynth is a company creating a plug & play device that allows for streamlined monitoring of IoT devices. This company provides a way to monitor machine functionality and anomaly detection which could be attributed to cyber-attacks. The main advantage this company offers is the ease of use with a simple plug & play device that can be connected to a vast number of IoT devices (*Production Monitoring*, 2022). Our goal is to create a solution similar to this but geared specifically toward our problem space.[7]

Network-based Cybersecurity Solutions

In operational technology facilities, there are many current solutions to both improving interconnectivity and efficiency of operations as well as cyber security for computers inside the facility networks. There are several naive methods of network security currently in practice. Oftentimes, there is a "firewall" in place to protect against outside bad actors. "Firewall" usually is a gross generalization of any technology that secures the network from unauthorized access, such as from unauthorized IPs and through SSH access, and can range widely in levels of security.

Virtual Private Networks (VPNs) are a more specific way of allowing secure remote access to a network. VPNs help devices stay private online and securely connect a user to an internal network by rerouting one's network traffic through a VPN proxy server (Empey & Latto, 2022). This proxy server acts as an intermediary between the user and the targeted network, adding proper encryption throughout the process. Another solution to help strengthen firewalls is the (software-defined perimeter) SDP (*What Is a Software-Defined Perimeter? | SDP Vs. VPN*, n.d.). An SDP verifies incoming users through transport layer security and hides a network from external parties. This system (along with many others) implements a practice called Zero-Trust Security, where a system assumes that no user, device, or network is considered trustworthy by default, requiring verification at every step. However, according to the interview with Randy Marchany, many IoT devices still have their default passwords set, which in turn nullifies a lot of these verification steps.[8]

Air Gap

Network segmentation and "airgapping" is another way that OT facilities secure their networks. Network segmentation is a method to control traffic flow (*What Is Network Segmentation?*, n.d.). This is beneficial as it can limit cyber-attack ranges and the scope of compliance for certain areas, reducing cost and overhead. Airgapping is the practice of effectively cutting a device or a network away from the internet; this is the most secure "firewall" as it requires physical access to the intranet/device to interact with it. Many legacy systems are airgapped and require physical interfacing and USB drives to update the firmware or reprogram PLCs. However, according to an interview with Monta Elkins from the SANS

[7] Zerynth - Soham
[8] Network-based Cybersecurity Solutions - Forrest

Institute, he says that it is almost impossible to have a purely airgapped network as long as there is one device that has some sort of internet connection.[9]

Cloud

Currently one of the major systems utilized by companies to manage their network and information is cloud architecture. There are multiple different cloud setups each providing different benefits that help to protect the security of the company. The main solutions today are cloud providers, co-locations, and private internal cloud networks.

The first solution, cloud providers, is considered the most widely used in the industry today. One of the big benefits cloud providers provide is a way to easily utilize their networks without the need to support your infrastructure or software to develop on top of it. Today there are numerous different cloud providers with many providing specific solutions for IoT devices. Companies include AWS IoT, Azure IoT, IBM IoT, and AT&T IoT all of which are solutions that are specific to IoT devices to help provide an increased level of security. These solutions typically provide more security options and features tailored to the IoT devices which tend to be more vulnerable due to the low-compute and cost nature of the devices.

The second solution, co-locations, is a hybrid between cloud providers and private internal cloud networks. Co-locations allow a company to host their cloud network offsite in another company's facility. This allows a company to host their private servers without having to maintain or upkeep the servers and resources involved with hosting servers. This option allows companies to create their custom security solution which leads to a large range of overall security of the network based on the company's settings.

The third solution, private internal cloud networks, tends to provide the most false security when it comes to security since many of these networks are considered private or air-gapped but in reality, nearly no system is truly air-gapped. Based on an interview with Monta Elkins we were able to discover there are no true air-gapped solutions since nearly every network has one port or gateway that connects the network to the internet. This in essence tends to be one of the least secure options especially when implemented incorrectly.[10]

Compliance with Security Protocols

There are a lot of cybersecurity regulations and protocols that can cost a lot of money to get licensed and meet. Some of the regulations include the International Electro-technical Commission IEC 62445 (electronic security across several industry sectors), IEC 62443-3-3 regulation for secure industrial autonomous and control systems, NISTIR7628 for smart grid cybersecurity, NIST SP800-183 (which focuses on the network of things), NIST SP 800-82 (focusing on ICS Security), and CMMC, which is the maturity level of cybersecurity implementation in a system (Ahmadi-Assalemi & Al-Khateeb, n.d., #).

[9] Airgap - Forrest
[10] Communication - Soham

NIST SP800 and CMMC are most prevalent in evaluating cybersecurity generally in the United States, while the other standards are more specific to industrial applications.

According to an interview with Adam Rossi, founder of TotalShield and who is currently running a manufacturing business for defense applications, companies have a decision to make: whether to keep with old practices of manually looking at network logs and telemetry data to monitor processes, or spending capital to connect devices to a monitorable network system. It takes around $5,000-$15,000 to run a NIST SP 800 audit and can cost a company between $35,000 and $115,000 to fix any violations (*A Complete Guide to NIST Cybersecurity Framework*, n.d.). Similarly, CMMC 2.0 compliance can range between $30,000 to $200,000 (*CMMC 2.0 Certification Costs*, 2022). For any government contractor, compliance with NIST SP 800 and CMMC is necessary, and fulfilling that requirement can take a significant bite out of their allocated budgets.

From interviews with Adam Rossi, Randy Marchany, and other individuals working in the industrial cybersecurity space, a common trend observed is that manufacturers and OT have a belief of "if it works, don't fix it." This is in part because many OT systems cannot risk downtime, as it could either lead to a loss in productivity in a manufacturing setting or can cause temporary resource outages in power systems settings. This leads to many systems being extremely outdated, such as old versions of SCADA and monitoring software being updated fifteen years after they were initially installed, as well as new technology required to retrofit legacy systems. Since these facilities have a belief that since they have not been hacked yet, it won't happen, many continue to keep bad practices which increases their attack surface and vulnerabilities, leading to possibly devastating costs in damages and recovery.[11]

Statistics

Many quantitative statistics support previous claims that the manufacturing industry is at a big risk of more prevalent and effective cyber-attacks.

According to Contrary Research's report on the cybersecurity landscape, 98% of IT and security leaders said they dealt with at least one cyber-attack in the past year, and that for CIOs and CTOs, cybersecurity is one of the priorities in digital transformation business plans. As more people are online and devices are interconnected, the attack surface grows tremendously, resulting in an increase in spending for risk management (11% increase) and cyberattack costs to $10.5 trillion by 2025 (Odum, 2022).

IBM's X-Force Threat Intelligence Index for 2022 marked manufacturing as the most attacked industry in the world, with 47% of the entryways through vulnerabilities and exploits (*IBM Security X-Force Threat Intelligence Index*, n.d.). There also was a 3000% surge in IoT malware activity between 2019 and 2020 as attackers start to blend their traffic with cloud and network activity. IBM reported that there was a 2204% increase in adversarial surveillance on popular SCADA protocols between January and September of 2021, as the vulnerabilities related to IoT devices have increased 16% year over year; for ICS systems specifically, the growth was 50%. Even as facilities move towards the cloud, attackers follow by attacking Linux environments with a 146% increase in Linux ransomware and Docker-focused targeting (*IBM*

[11] Compliance with Security Protocols - Forrest

Report: Manufacturing Felt Brunt of Cyberattacks in 2021 as Supply Chain Woes Grew, 2022). Another IBM X-Force report mentioned that in Europe, Asia, and the Middle East, 50% of attacks were caused by unpatched vulnerabilities and that patching these vulnerabilities is the biggest struggle for many companies.

Deloitte's report on cyber risk in advanced manufacturing shows that 31% of enterprises have not conducted an ICS assessment, even though 35-45% use smart sensors and products (*Cyber Risk in Advanced Manufacturing*, n.d.). 50% of the companies have performed an ICS vulnerability assessment less than once a month, 39% have recognized a breach in the past year, and 38% had losses between $1-10 million. Many companies have chosen to airgap or segment their networks; however, Deloitte reported that this could have downsides to companies improving their technical applications. For companies that do have connected devices (around 50%), 76% of the companies chose to use Wi-Fi protocols. As many of these attacks (such as REvil) come from mostly human mistakes in phishing attacks, it is important to provide extra security gates to

The addressable market size for IoT security is currently around $100-200 billion, with only around a 5% penetration (Aiyer et al., 2022). According to McKinsey, closing the log/activity visibility gap is a priority with CISOs. Additionally, according to Cisco, crypto mining is one of the biggest threats overall and takes up over three times as many endpoints in manufacturing than ransomware - even though ransomware is the most prevalent in manufacturing compared to other industries (*2021 Cyber Security Threat Trends-Phishing, Crypto Top the List*, n.d.). According to TrendMicro, the average financial cost of a cyber-attack in OT costs around $2.8 million and 40% of companies were not able to prevent the initial attack (*Cyber-Attacks on Industrial Assets Cost Firms Millions*, 2022). This shows that it is necessary for extra gates in security for OT facilities, so that ransomware, botnets, and similar attacks do not exploit vulnerabilities and threaten the efficiency and capital of more OT facilities.[12]

Bounding Constraints

Scope

By focusing on IoT devices connected to SCADA systems, especially large actuators, companies can create a safe and reliable solution to validate firmware updates. The use of new technologies such as hashing, blockchains, and TEE technology can help ensure the security and correct operation of these devices, reduce the risk of cyber-attacks, and improve the efficiency of industrial operations[13]. Furthermore we are focusing on mid-sized manufacturers as the first application space for our system.

Function

The function bounding constraint for this focus area is based on streamlining IoT firmware updates, while securing the update itself via technologies rather than having the individual physically installing the

[12] Statistics - Forrest
[13] Scope - Somya

update being the sole line of authentication[14].

Temporality

Our solution concept is informed by a variety of requirements; one of them being an implicit statement that update length should be at industry standard. The range of a few minutes to a few hours is a reasonable target time for our update length and meets an industry standard[15].

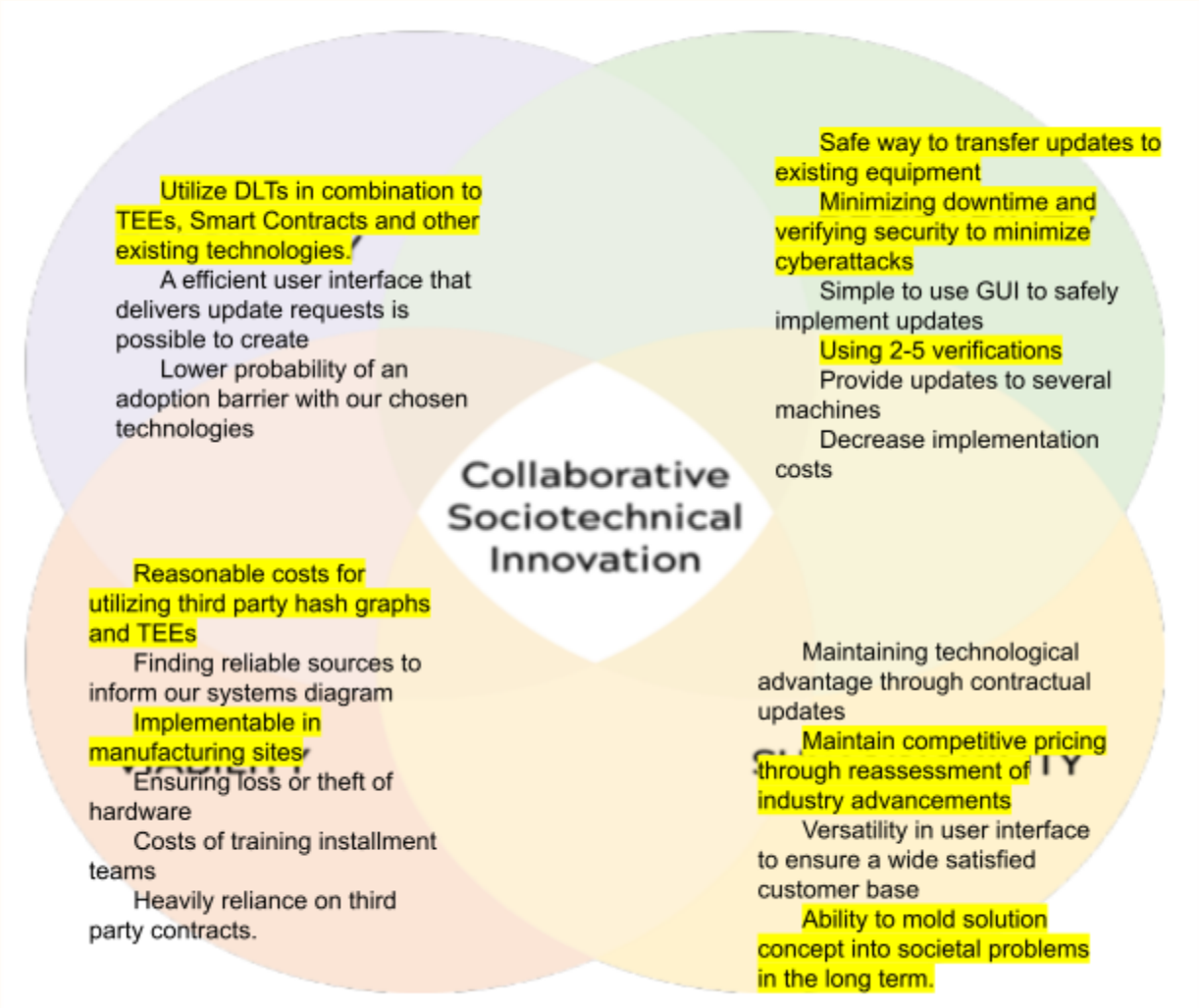
Composition

There are several components to our solution concept considering that it is a multi-step verification process. There are two main sides to the solution concept, one being OEM, the other being manufacturing IoT devices.

The OEM portion of the system will have file receiving and encryption technology through an OEM ledger and estuary. This will require a Smart Contract to add and send updates to the receiving Manufacturing IoT Device. The potential solution will require the use of blockchain/hash graph technology to peer-to-peer verify the transaction of data. Transfer of data will require the use of FileCointo store the data and Firebase cloud services. A final check will be achieved through the use of TEEs. See Systems Diagram for a detailed description of the interconnectedness of the pieces of composition.[16]

[14] Function - Diya and Priya
[15] Temporality - Rakesh
[16] Composition - Diya

Four Set Template



[17]

[17] Four-Set Template - Isabella

Solution Concept Analysis

Desirability

Multiple desirability attributes are identified with rationale and details of alignment to stakeholder needs.[18]

Stakeholder Alignment and Requirements Decomposition

This table visualizes the main stakeholder and user requirements and desires we have determined through research and interviews. The table also displays derived requirements discovered from user and stakeholder inputs.

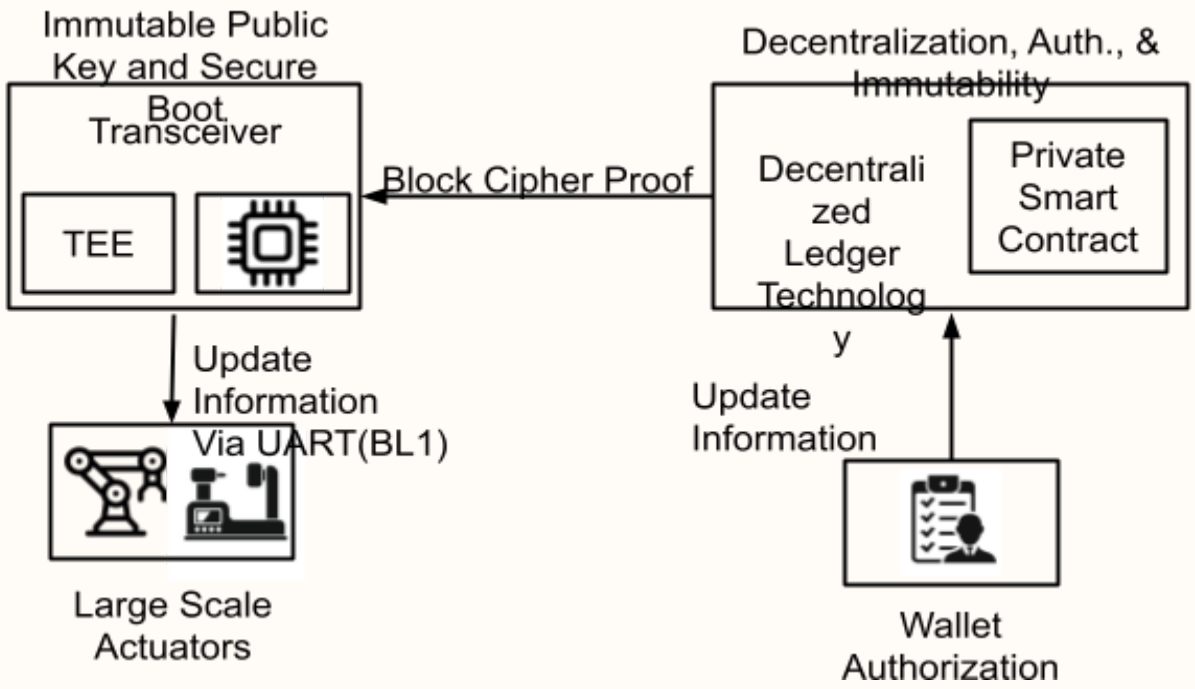
Source	Category	Need	Comment
Stakeholder	Equipment Suppliers	(1) Need a safe way to transfer updates to the manufacturers utilizing their equipment	Rockwell, Honeywell, ABB, Stratsys
Stakeholder	Manufacturers	(2) Need a safe and streamlined way to update their machines on the manufacturing line without a significant amount of down time. The updates also need to be verified and trusted to ensure the possibility of cyberattacks is minimized.	MOOG, Boeing
User	IT Professionals managing updates in listed manufacturers	(3) Need a simple to use GUI that enables the safely and securely send and implement updates to machines	
Derived		(4) Utilize 2-5 verification and security platforms as safeguards for one another.	
Derived		(5) Need to provide updates to at least 85% of machines at once to ensure that costs of downtime are limited.	
Derived		(6) Needs to be easily implemented to avoid overwhelming costs of implementation through downtime, cost of installation, and cost of consulting. For example, the average automotive manufacturer loses **$22,000 per minute when the production line stops - Forbes**	

[19]

[18] Desirability - - Isabella, Priya, Diya, Somya

[19] Stakeholder Alignment and Requirements Decomposition - Isabella, Priya, Diya, Somya

CONOPs



Above is a diagram that helps to describe how the system would operate from a high-level perspective. The diagram highlights the flow of information and the type of information that would be sent to the device at each step along the way.

To help better understand the diagram above, here is a sample user scenario that portrays how the information would be sent from the user to the large-scale actuators. First user A logs into their wallet and uploads the software through their wallet to a smart contract. This process digitally signs the information and sends it to the decentralized ledger which propagates the information to all the nodes on the network to verify the transaction. Once the transaction is verified the update is held in the network until the transceiver calls the smart contract using its key stored in the trusted execution environment. The transceiver sees there is an update for the large-scale actuators and downloads the information. This would be the step at which the information would be decrypted if the user elected to encrypt the update. Once decrypted the transceiver would interact with the large-scale actuator and update the firmware. This process involves minimal involvement from user A and ensures the update given to the large-scale actuator is the update user A intended the large-scale actuator to receive.[20]

MIRs

Requirement	Implicit/Explicit	Qualitative/Quantitative	MIR

[20] CONOPs - Soham

The process of setting up the device and getting it started needs to be simple. The technology itself needs to be non-invasive.	This is an implicit requirement since our solution concept's strengths are the decentralized updates and their ease of operation.	This is a qualitative requirement as there are no numbers attached or measurements to be made. This requirement expresses a desired outcome of the stakeholders in this problem space.	While this is an important requirement, it would not be considered the most important since other requirements take precedence.
Updates need to be finished within certain periods of time. Small software patches must be under 2 minutes, and larger updates need to be under an hour. Polestar, a company that builds 100% electric cars can take up to approximately 90 minutes for their over-the-air firmware updates to their cars (*Over-The-Air Software Updates (OTA)*, n.d.). On the other hand, Ford, a company that builds hybrids and less electronically intensive cars, takes approximately 3 minutes for their small software updates for their hybrid and electric models (Maciuca, n.d.).	This is an implicit requirement that is based on the industry standards and it is derived from our solution concept being a quicker solution than what is available today.	This is a quantitative requirement since it dictates the time constraints on our updates.	This is not an important requirement as our solution concept's most important goal is security. While speed is important, there is no explicit requirement from a stakeholder stating that updates must be sent and received in a certain time period. However, the updates still need to be sent in a timely and appropriate manner.
It is crucial that our	This is an explicit	This can be considered a	This requirement is an

solution concept is able to meet regulation standards for the NIST SP800 and CMMC 2.0.	requirement as these rules and regulations are an industry standard and are required for any implementation of a solution concept in this problem space.	quantitative requirement as there are many requirements within the standards and many of those requirements are numbers and measurements based.	MIR because in order for our solution concept to be implemented in the real-world, it needs to follow real-world regulations and protocols. These governmental regulations and cybersecurity compliance rules will play an intensive role in shaping our solution concept.
Updates must remain secure and unauthorized devices should not have access to any private/secure information.	This is an explicit requirement as it is the basis for our problem space and a fundamental requirement for the stakeholders.	This is a qualitative requirement as there are no numbers involved and the requirement expresses a desired outcome from the stakeholder.	This is an MIR because it is stated from a stakeholder's desirability need, it impacts each of the sets, and presents different causal links. For instance, one causal link is that the stakeholders desire for fast and secure updates informs the viability of the solution since faster and more secure solution concepts would require more advanced technology.[21]

Through Stakeholder Alignment and Requirements Decomposition we discovered three Most Important Requirements. These requirements are Security, User Friendliness, and Efficiency.

As for the first MIR, Security, we learned that customers' main requirement is that the new system provides a secure means of updating legacy devices. Current legacy devices and systems are all interconnected through ethernet connection on a closed environment. Relating to the Efficiency MIR, as companies move towards over-the-air streamlined updating systems, they are faced with new Cyber

[21] Risk Table - Rakesh

Security challenges, and have many more weaknesses. Adopting a newer, technologically advanced system would only be beneficial if the passage of data remains secure.

The second MIR is being User Friendly. This means that the customer will be able to interpret and interact with the solution concept effectively and efficiently. This will ultimately enable them to use it. Simplicity in installment will allow users to understand the components. It is required that the user interface used for receiving and transmitting is easily understandable.

The third MIR is efficiency. There is a need for streamlined, systematic updates for manufacturers that have several of the same machine, in order to avoid downtime costs. It is required that delivering updates over potential solution systems happens at an appropriate speed, and directly delivers updates safely as they are received. While security is also an MIR, if the security components thwart efficiency and don't eliminate downtime costs, the MIR of efficiency isn't met.[22]

Feasibility

Multiple Feasibility attributes are identified below.

Features & Benefits[23]

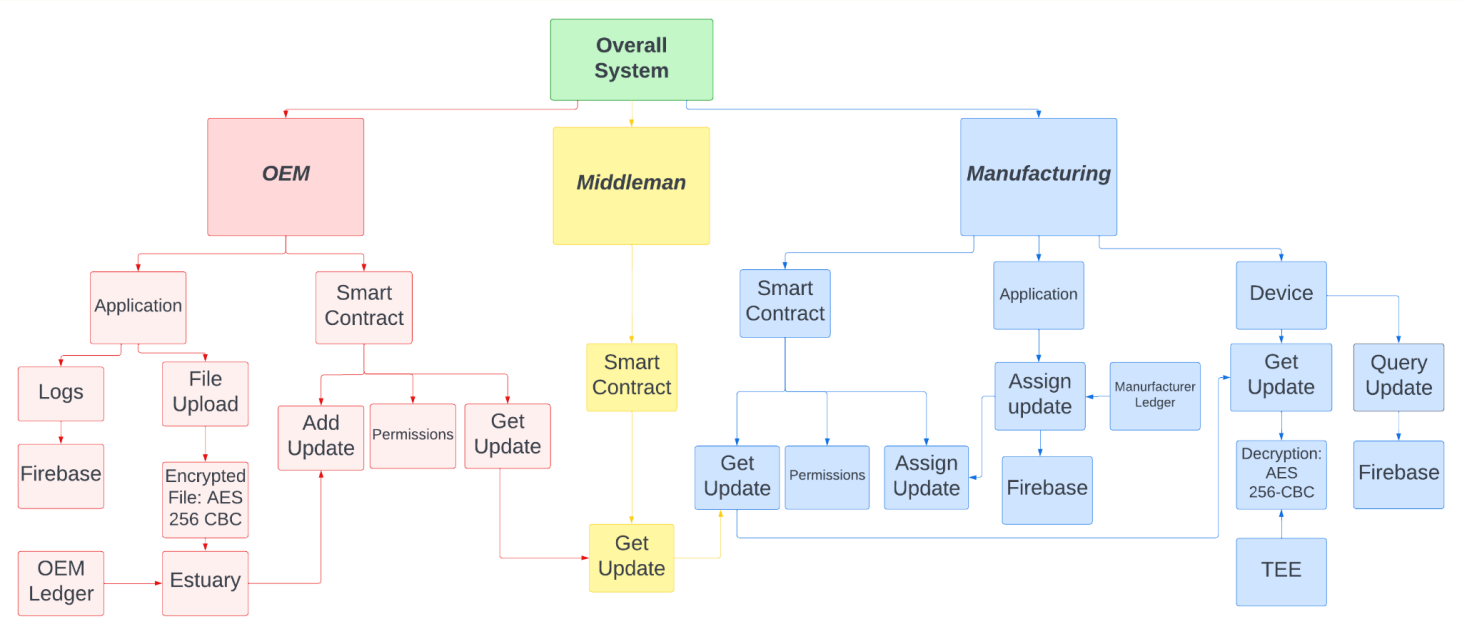
This table shows the main features determined from feasibility attributes, and which desirability attributes they are sourced from. The majority of the feasibility features are existent technology, proven to have worked towards security efforts. They all provide benefits to the goals of security and efficiency.

Source	Category	Attribute to Consider	Comment/Artifact
Desirability	Stakeholders (D4)	(1) Decentralized Ledger Technology (DLT) like hashgraphs enable transactions to take place in a decentralized, secure environment. Combining this with TEEs (which provide hardware-based protection from potential interference) ensures that firmware updates from the supplier will not be tampered with.	Constraint
Desirability	Equipment Suppliers(1), Derived (4)	(2) TEE's are small eFuses, and built-into the main processor of the IoT Device.	Specifications/Constraints
Desirability	Stakeholders	(3) System Diagram would include: user interface that delivers requested updates, DLT to verify transactions, manage updates & store updates, TEEs for trusted execution of code based on hardware for secure boot & manage OS.	System Diagram, Functional/Physical Decomposition

[22] MIRs- Isabella, Rakesh
[23] Features and Benefits - Isabella, Priya, Diya, Somya

Desirability	Stakeholder & User (D6)	(4)Deloitte's report on cyber risk in advanced manufacturing displayed that 31% of firms have not conducted ICS assessments even though 35-45% of the industry use smart products. This is most likely due to the adoption barrier attributed with implementation of such technologies.	Constraint
Desirability	Derived (D5)	(5)A user interface to allow updates to be sent remotely without human interference for each individual machine	Physical/Functional Decomposition

Systems Diagram



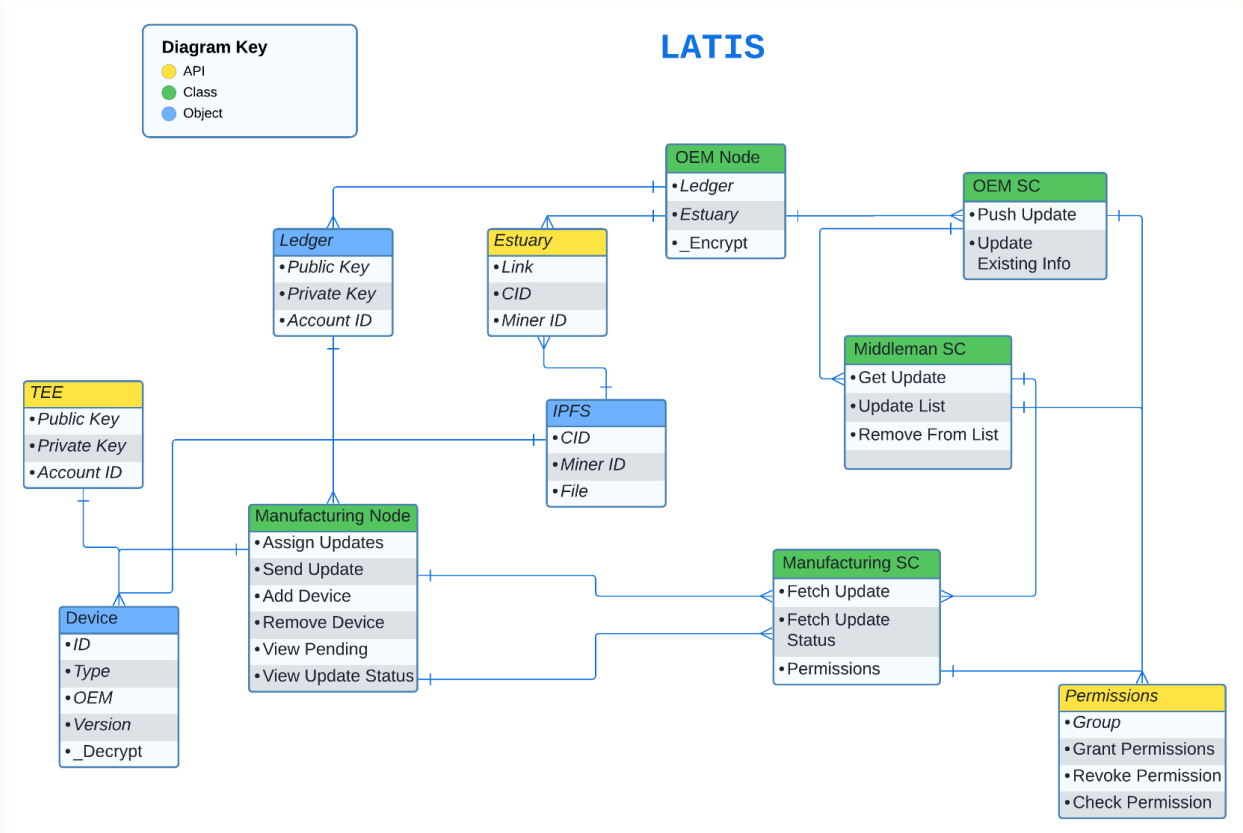
[24]

Above is a high level systems diagram that explains the flow of information from the OEM who supplies the machinery to the manufacturing plant where the information is used to update devices. In terms of functions; as seen from the chart above information is uploaded by the OEM at which point the file is encrypted to ensure file security. The private information of files such as checksums, salt, and location are sent to the smart contract to be stored while the file itself is sent to Estuary to be stored on Filecoin through IPFS. At the same time information is populated in Firebase cloud service explaining the update type and indicating any information relating to the update. Once the update is sent to the smart contract for the OEM it is transferred immediately to the middle man contract informing the manufacturer of a pending update. This update is then fetched by the manufacturer who assigns the update to a specific device. The device receives a ping from the Firebase database stating it has a pending update and

[24] Overview System Diagram: Content - Soham, Formatting Fille

requests the update from the smart contract. The update is downloaded and decrypted based on the information provided and then sent to the actuator through a secure UART protocol.

The form this system takes will be a user interface, information sent over-the-air, and the receiving machine.[25]



[26]

The above diagram is an in-depth view of the functionality each aspect of the system has to offer. The diagram helps to highlight the complex nature of the contracts and the way the contracts interact with each other to create a secure system ensuring inherent trust and security.[27]

Technology Readiness Level

Distributed ledger technology (DLT) has only been around for a few decades, but has already been implemented in many different industries. Due to the inherent trust in DLT, it has already been successfully integrated in different parts of the manufacturing industry. For example, multinational automotive corporation, Daimler AG, relies on Marco Polo (a blockchain platform to manage transactions) to secure and track payments.

---

[25] System Diagram Description - Soham
[26] Software Diagram - Rakesh
[27] Software Diagram Description - Soham

The potential applications of DLT in industry is only increasing. In the past decade, industrial manufacturing has been moving towards intelligent manufacturing plants by becoming more digital and autonomous. With customers desiring individualized products and manufacturers constantly having to adjust to market demands, enabling flexibility is a key goal for many manufacturers. Manufacturing-as-a-service can realize these goals through manufacturing plants that are used by different tenants. DLT is a viable option to implement manufacturing-as-a-service by enabling decentralized trust and traceability. As a key player in industry 4.0, DLT will continue to disrupt and transform industries and specifically, the industrial and manufacturing sector.

Our prototype is at a TRL 3 since we have so far provided a proof of concept. It is yet to be validated in a lab or relevant environment. However, there is great potential for our solution concept and the prototype itself demonstrates many of the qualities that original equipment manufacturers find attractive.[28]

Viability

Below is a description of Viability attributes of our solution concept. The table references the tables found in Stakeholder Alignment and Requirements Decomposition and Features and Benefits. The table also sources which desirability or feasibility attribute the viability attribute is derived from. The main focus of our attribute table is which costs related to implementation and hardware/software are important to consider. [29]

Source	Category	Attribute to Consider	Comment/Artifact
Feasibility and Desirability	F(1) Stakeholder(D4)	Cost of utilizing third party hash graphs and file-coins to their secure environment.	Business Case/Systems Diagram /Functional and Physical decomposition
Feasibility and Desirability	F(2), Equipment Suppliers(D2), Derived (D4).	Purchasing and implementation cost of TEEs.	Business Case/Systems Diagram /Functional and Physical decomposition
Feasibility	F(3)	Finding reliable sources of manufacturing implementation of our systems diagram.	Systems Diagram
Feasibility and Desirability	Derived(D6) Stakeholder(D2) (F3)	Can it be implemented at different manufacturing sites with efficiency.	Analysis
Feasibility	(F1),(F2),(F3)	Ensuring our hardware pieces aren't lost or misplaced by the company post-implementation.	Systems diagram and physical decomposition
Feasibility	(F4) (F5)	Cost of training installment, security, and	Business Case

[28] Technology Readiness Level - Rakesh
[29] Viability - Isabella

		admin teams	
Feasibility	F1, F3, F5	We rely heavily on third parties like FileCoin and Hadera, and Smart Contracts, which can accumulate costs.	Analysis

Design

In terms of design, we are using the Agile method to perform sprints on developing and iterating on key features of the system. There are several aspects we need to consider designing for: the interface for the users to interact with, the secure file and data transfer and upload, and secure download and installation of the firmware update.

We decided to use Filecoin to serve as the IPFS uploading and hosting service, as well as Estuary for the Node based client. We decided to use SvelteKit for the application stack. Since we will need to use backend APIs, we decided that building our own application would be the best for maximum flexibility.

For the DLT based contract execution, for speed and cost purposes, we decided to use Hedera as it is a hashgraph technology that does not require extreme amounts of compute for consensus. This allows the possibility to host it on a local, internal network.

Finally, for the firmware update concept, we can use a PXE boot for simple proof of concept, since we can utilize ARM based bootloading sequences to input and upload firmware updated data through an external UART or USB connection.

Testing

Analysis

Since our system is fairly complex, we plan on analyzing the viability and effectiveness of our solution by both historical evidence from other projects and businesses which approach the problem in a similar way as well as testing different things in small components and isolated environments.

For looking at other examples of work, we talked to Al Salour of the St. Louis Boeing facility, who led the trial projects with SIMBA Chain, a no-code solution built on Ethereum as a DLT solution to supply chain management and tracking optimization. By sharing data about the supply chain through SIMBA as an intermediary, they could track more information throughout the supply chain and know that the information could be trusted, shared, immutable, and secured, especially as the ownership of the data changes hands. This showed the value add of having a DLT in place for data sharing and transmission, as compared to email or physical spreadsheets. Since SIMBA is an L2 chain, it is comparable to Hedera. We chose to select Hedera as according to benchmarks online, it is more suitable for our purposes of local hosting networks, fast and cheap transactions for possible telemetry, and that Boeing also has involvement with the Hedera group.

We also analyzed other companies focusing on OTA updates, such as Zerynth, which developed add-on devices and RTUs to any industrial machine, facilitating OTA telemetry and monitoring. This example

showed the demand for modular ways of augmenting machines to the internet for updates and monitoring.

We first analyzed our solution by drawing out concept solution diagrams multiple times with iteration and on different levels of abstraction on whiteboards. This allowed us to figure out what is needed to solve the desired problems and determine what to

To test the feasibility of our prototype solution, we use a mix of small tests with quickly prototypable solutions as well as past experience.

Since Soham and Forrest have experience in PXE booting, bootloader architecture, and full stack application design, with projects to show for it, they believe that they do not need to test the viability of using these as a part of the prototype due to past experience and familiarity with the technology. We were able to build a simple prototype mockup for the clients within a few minutes.

To test the Hedera network, we used the Hedera testnet to see if the contracts we could deploy actually perform like we expect them to. Since it is on a testnet, we are able to deploy as many contracts as needed without spending any gas fees (The intrinsic gas cost is 21,000 per transaction plus the cost of input data (16 gas per non-zero byte and 4 gas per zero byte).

To test the Filecoin network, we could use Estuary, a free client service that easily stages and pushes files onto IPFS without the complexity of finding miners and using CID. It also provides information about the file back to us, so we can evaluate the effectiveness of the API.

To evaluate the effectiveness of the AES-256-CBC encryption we plan on using, we use checksums to compare file integrity after decryption and use online benchmarks to decide on IV, salt, and key lengths.

To analyze the speed of programs, we plan on using software based timers to figure out the speeds of completion for a task, comparing them to acceptable times specified by industry standards, stakeholder desires, and our own requirements. This is harder for the PXE boot unless we use hardware timers, so we will just generally time it using an external device (up to 45 minutes and sometimes more than one hour).

Demonstration

In our prototype demonstration, we will have a full end to end working system where:

- The OEM can upload a file greater than 400MB to the application, write logs and other specifications, encrypt it, and upload it to Filecoin within 2 minutes
- This automatically is validated with a Hedera contract and updates the MiddleMan contract for the Manufacturer
- The Manufacturer automatically gets a Firebase notification that there is a new update available. By manually refreshing the client, they can pull information from the MiddleMan contract about the new update.

- The Manufacturer can send the data through Hedera to the IoT device (RPi 3b), which it would then sign off on the transaction and pull the data from IPFS. It should then be able to decrypt the file within 10 minutes.
- This device should then be able to automatically PXE boot another computing device (RPi 3b)

This prototype system should work mostly automated and should have full transparency of information between the OEM and the Manufacturer. No party aside from the network Admin should be able to access the Middle Man. All parties must have a Ledger hardware wallet to certify their authority. For demonstration purposes, more data than an actual system would show could be output to a continuous terminal.

Inspection

For inspection on the overall security of the system we plan on utilizing skills heavily used in the field of chaos engineering. Through these techniques we will insert different malware and cyberattacks in a planned manner to minimize damage while validating the integrity of the overall system.

The system will be able to be inspected by determining file integrity through checksums conducted at the sender side and the receiver side to verify no malicious activity as well as reporting logs to detect if any attackers were able to access the encrypted data during transit or storage.

The final aspect for inspection will be the actual inspection of the update ensuring the device had the proper update installed and all systems operating as they were previously validating the update wasn't corrupted.

Test

For testing, we aim to create a prototype in the future to be piloted at the Learning Factory on campus in Durham Hall. This is a great space that imitates an Industry 4.0 environment, so we can test our idea and modify it accordingly so it can be applied to other similar environments. As Matt Earnest has told us, he is aiming to have an open network between labs; DLTs could provide a value add of trust and authority between different parties on campus, also ensuring that there are no outside bad actors who could be an imposter in the system.

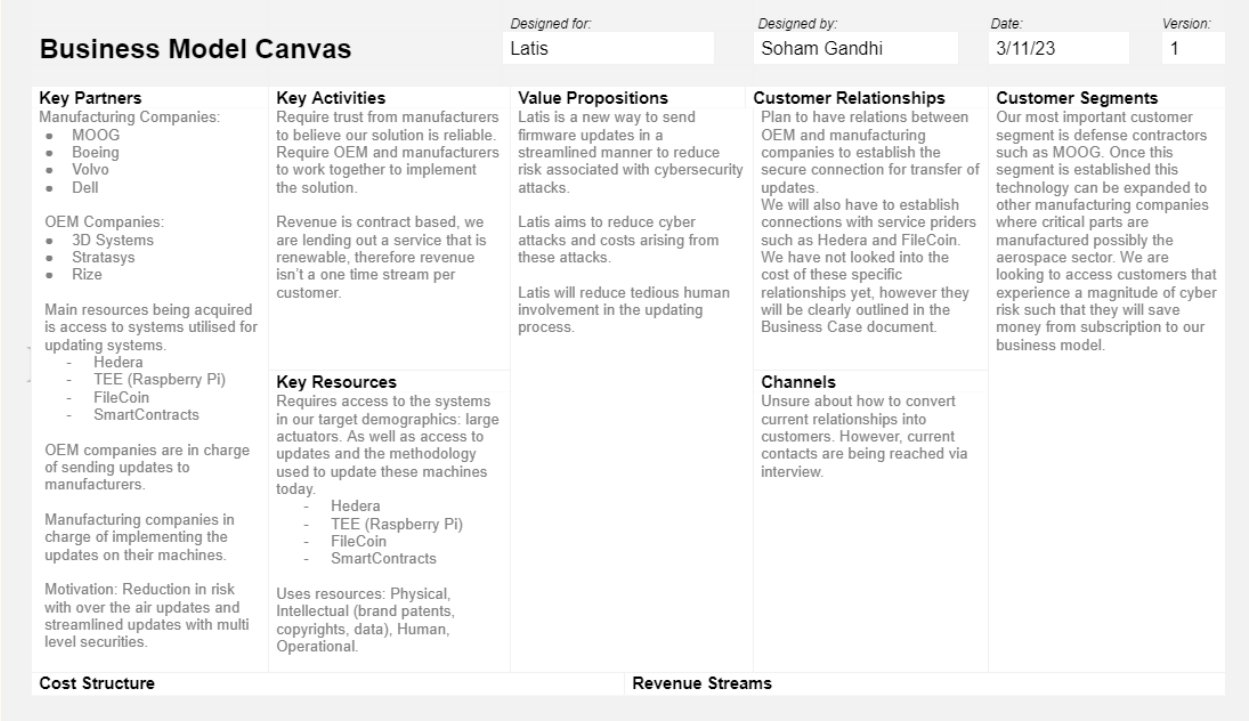
Currently, we will just be testing our system on our own and school networks, with possibly VPNs to simulate a private network which is still connected to the internet.[30]

Tests will include our updates speed which is a main requirement by stakeholders to eliminate downtime. We will be keeping it below industry standard which again is up to 45 minutes and sometimes more than one hour. Another test will be if we could connect at least 100 devices, which is the minimum manufacturers normally have.

[30] Testing - Forrest, Soham, Isabella

Business Case

Latis's primary value proposition with regards to facilitating secure firmware updating is to reduce the frequency of cyber attacks in the manufacturing industry. Common types of cyberattacks that the manufacturing industry faces include Ransomware, Intellectual Property Theft, and IoT Attacks such as malware installation. These attacks, apart from costing a company money to respond to, also cost a great deal in recovery. A company's costs include a loss of customers, declining stock prices, potential loss of IP, reputation damage, and even insurance premium increases of up to 200%. IBM Security and Ponemon Institute found that the average cost of a data breach for a manufacturing company was $5.07 million USD. Moreover, According to MAPI, 40 percent of manufacturing firms experienced a cyber attack in last one year. Jointly, these statistics express the commonality and impact of cyber attacks to manufacturers. Latis' value proposition is described in more detail in the following business canvas.



Business Model Canvas

Designed for:	Designed by:	Date:	Version:
Latis	Soham Gandhi	3/11/23	1

Key Partners

Manufacturing Companies:
- MOOG
- Boeing
- Volvo
- Dell

OEM Companies:
- 3D Systems
- Stratasys
- Rize

Main resources being acquired is access to systems utilised for updating systems.
- Hedera
- TEE (Raspberry Pi)
- FileCoin
- SmartContracts

OEM companies are in charge of sending updates to manufacturers.

Manufacturing companies in charge of implementing the updates on their machines.

Motivation: Reduction in risk with over the air updates and streamlined updates with multi level securities.

Key Activities

Require trust from manufacturers to believe our solution is reliable. Require OEM and manufacturers to work together to implement the solution.

Revenue is contract based, we are lending out a service that is renewable, therefore revenue isn't a one time stream per customer.

Key Resources

Requires access to the systems in our target demographics: large actuators. As well as access to updates and the methodology used to update these machines today.
- Hedera
- TEE (Raspberry Pi)
- FileCoin
- SmartContracts

Uses resources: Physical, Intellectual (brand patents, copyrights, data), Human, Operational.

Value Propositions

Latis is a new way to send firmware updates in a streamlined manner to reduce risk associated with cybersecurity attacks.

Latis aims to reduce cyber attacks and costs arising from these attacks.

Latis will reduce tedious human involvement in the updating process.

Customer Relationships

Plan to have relations between OEM and manufacturing companies to establish the secure connection for transfer of updates.
We will also have to establish connections with service priders such as Hedera and FileCoin.
We have not looked into the cost of these specific relationships yet, however they will be clearly outlined in the Business Case document.

Channels

Unsure about how to convert current relationships into customers. However, current contacts are being reached via interview.

Customer Segments

Our most important customer segment is defense contractors such as MOOG. Once this segment is established this technology can be expanded to other manufacturing companies where critical parts are manufactured possibly the aerospace sector. We are looking to access customers that experience a magnitude of cyber risk such that they will save money from subscription to our business model.

Cost Structure

Software: The most expensive aspect of this system would be the transfer of information and the initial cost to deploy contracts to the network. There are minimal costs to maintain the contracts and transactions. These software costs include accessing systems used in the process of creating and transferring updates. Such services are detailed in they key partners section of this Business Canvas (ex. Hedera).

Hardware: From the hardware side the cost is finite and upfront. The main cost on this portion is with the development of the hardware and possibly the custom PCB.

Latis is a value driven business as it's aim is to provide a service that will ultimately reduce cyber-risk.

Revenue Streams

The model we plan to utilize is infrastructure as a service. There will be a small cost at the beginning to pay for the initial cost associated with the hardware followed by a monthly fee for maintenance of the smart contracts and continual cost associated with transactions.

Another model could be based on transaction with an initial fee to pay for the startup cost. This model allows for the company to choose how much they update and allows them to pay as they go rather than a fixed monthly cost.

Both of these models, however, will likely have a fixed-pricing model, rather than allowing for regular price fluctuation through negotiation or market changes.

As Latis aims to provide its services to mid-sized manufacturing companies, a critical aim of this business case is to discern the approximate value Latis can provide such manufacturers by protecting them from

the cyberthreats expressed above. The mid-sized manufacturer upon which we will be centering this case study is MOOG, a manufacturer located in Blacksburg, VA. We have decided to focus on mid-sized companies as most large manufacturing firms already have a system of their own and do not require a third party like Latis.

As such, for the sake of this case study, it will be critical for us to define our bounds of a 'mid-sized' manufacturing company. The manufacturing industry as a whole, has a market valuation of approximately $7.0T, based on it's annual revenue. This revenue is contributed to by a number of mid-size companies earning on average anywhere between $10 Million and $1 Billion, according to Indeed's definition of a mid-sized company. MOOG itself, makes an average of $3 Billion in revenue per year, putting it on the upper end of revenue in the mid-size market. As our solution concept will be tailored to MOOG for this case study, Latis will be defining it's target market of mid-sized manufacturing companies to be those earning between $50 Million and $5 Billion.

Our beachhead market within this larger target market of mid-sized manufacturers is defense-oriented mid-sized manufacturers. We have chosen this market because defense manufacturing, specifically due to it's proximity to government work, requires security. Thus, they will be the most likely to invest in their IOT device security to Latis. The approximate valuation of the defense-oriented mid-sized manufacturing industry can be expressed as the value of this industry's sales. Total defense sales in the US are approximatley $225.5 Billion. As a fraction of the total manufacturing industry, this makes up approximately 3.22%.

Moreover, capturing this beachhead market will allow Latis to better understand security needs in many other manufacturing markets, as the defense industry likely has the most standardized and extensive security protocols. These will likely be applicable to many other sectors. Of this beachhead market, Latis currently is working with 0 companies. However, Latis hypothesizes that it will be able to access 3 companies within the first 3 years of operation, given its current outreach and communication with clients. This would amount to approximately 4% of the beachhead market (assuming MOOG's average revenue).

MOOG is a manufacturer of Electro-Mechanical Motion Control Products, such as Motors, Alternators, Resolvers, and Utility Actuators. As MOOG has a significant number of government contracts, much of what they manufacture requires government-grade security, making MOOG a prime example of a potential Latis end-user. In total, Blacksburg MOOG houses around 200 IoT manufacturing devices. These include 3D printers receiving encrypted data, vibration sensors, extreme temperature machines, and space simulator machines. Most manufacturing devices used in research and development receive new updates frequently to stay up to date with new features. The release of firmware updates can range from monthly to once every multiple years. Thus, for the sake of this case, given MOOG's commitment to development, it can be assumed that a company sized similarly to MOOG will require approximately 200 firmware updates quarterly. This accumulates to a total of 800 updates annually. The size of this firmware update in Gigabytes is critical to determining the costs that Latis will have to incur to provide a service to MOOG.

As a service provider, Latis's profit is critically impacted by the service pricing strategy advertised. The two proposed pricing models Latis considered were a *general subscription model* and a *pay-per-update model*. In making our decision, we considered our customer's predicted usage habits/behaviors along with the financial desirability of each proposed model. We chose to continue with a general subscription model in order to financially appeal to our customers. For our case-specific predictions, we hypothesize that most IoT Manufacturing devices tend to update 4 times yearly. As such, a client interacts regularly with Latis for the majority of the business cycle. From a consumer standpoint, issuing a blanket payment rather than reaching out to Latis each time they would like to conduct an update provides enhanced cost effectiveness. This highlights yet another problem with the pay-per-update model. This type of model will require Latis to make a consistent effort to maintain contact with the customers so they do not forget about the service that we are offering them. A general subscription model appeases this concern by contractually obligating Latis to provide services for the client on a yearly basis.

Annual cost will be $50k flat. Alongside this cost, we will be creating a usage term agreement with each client. This contract will ensure that Latis will provide services to each client for period of 3 years to start.

Latis's costs can be broken down into variable, overhead, and NRE costs. Variable consists of Labor costs and Subscription Usage Costs. Labor costs refer to the monetary value of our team members' time and effort contribution. For the sake of this case study, these elements will refer to time and effort specifically directed toward MOOG as a client. Total variable costs per client will be monetized using the following formula: initial labor setup costs + subscription usage costs.

Overhead costs refer to those which will not change based on the number of clients being served. Physical overhead for Latis will be negligible as most operations will take place virtually.

Finally, NRE costs refer to one-time costs for product development. These include prototyping and concept development costs.

<u>VARIABLE COSTS:</u>

- **Labor Costs**:
    - There are a number of labor costs that Latis will be responsible for. These costs include all working hour costs besides working hours required for initial update setup.
    - Initial Update Setup Costs take into account hourly wage, hours required to set up updates, and the cost for a Raspberry Pi.
        - Initial setup cost: ($35/hr * 4hr/device) + $100/device = $240/device
- **Subscription Usage Costs**:
    - FileCoin
        - As of March 2023, the current market rate for storage on Filecoin ranges from around $0.07 to $0.10 per GB per month, but these rates can fluctuate over time.
    - Hedera
        - According to the Hedera website, the creation of a contract with default settings is $0.583.

- For each service offered by Latis, there are 3 contracts provided ranging in cost from 3 to 5 dollars.

Assuming quarterly cycle of updates per machine

TOTAL VARIABLE COSTS: initial labor setup costs + subscription usage costs.

Assumptions: This MOOG facility has 200 machines

Quarterly updates (4/year)

Total Number of Updates: 800 updates

Labor Costs = ($240/device) * (200 devices) = $48,000

Subsciption Usage Costs = $2/machine * (200 machines) * 4 = $1600

Total Variable Costs = $48,000 + ($1600 * 3 years) = $52,800

OVERHEAD COSTS:

- **No physical overhead costs**

NRE COSTS: Total NRE costs will be taken on by Latis, and will not be attributed to MOOG for the sake of this case study.

- **Prototyping Costs**
  - 2x Raspberry Pi 4 Model B with 8GB of RAM = $100 * 2 = $200.
  - Hedera cost - $60 for 900 tokens
  - Ledger cost (3 ledgers) = $80/ledger * 3 ledgers = $240
  - Contract Deployment and Updates = $12 + $1.99 = $13.99

    # of contracts * cost = 3 contracts * $4 average price = $12/contract deployment

    ($0.10/GB * 4.9 GB + $0.50 * 3 transaction calls) = $1.99/update

    **an update of 4.9GB was used as a placeholder for actual update size, given the comparability of a manufacturing update to a Windows update.

Total NRE Costs = $60 + $200 + $240 + $13.99 = $513.99

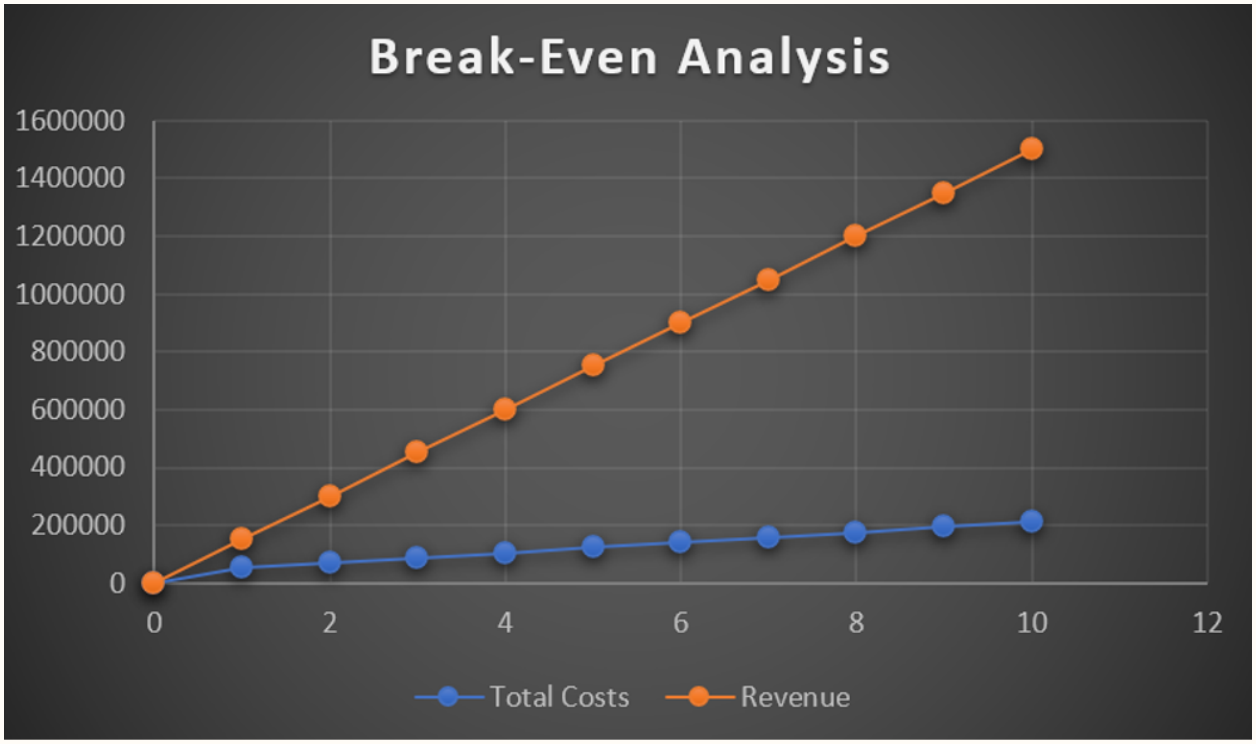
TOTAL COSTS attributed to MOOG for the first 3 years = **$52,800**

REVENUE = $50,000* 3 years = **$150,000**

PROFIT = **$97,200**

Given this cost and revenue analysis, it is clear that Latis will break even from any NRE and startup costs immediately upon acquisition of it's first mid-range manufacturing customer. This highlights the low-cost

business model that Latis has employed, which packages a number of services, technologically and financially, into one affordable end service.

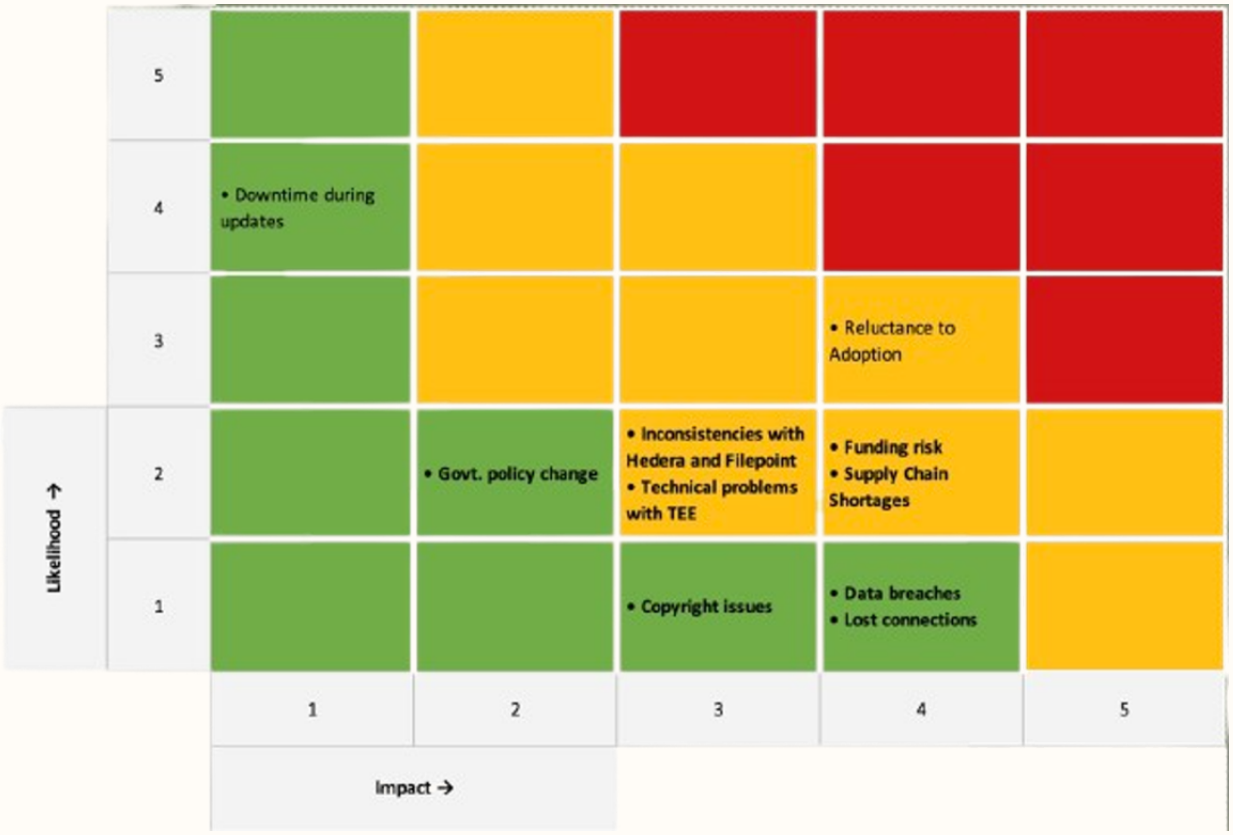


To highlight the monetary fulfillment of our value proposition, given the understanding that approximately 40% of manufacturing firms experience a cyber attack per year with an average cost of $5.07 million USD, we have determined that the cost of a singular round of 800 updates (annually) in a mid-sized manufacturing company using Latis is around $150,000.00. Given this pricing, MOOG could use Latis services for upwards of 100 business cycles before finally reaching the cost of an average cyber attack. This value is described in the graphic above. As such, the use of Latis as a risk management tool for manufacturing companies is heavily desirable.

A final aspect of the value offered by Latis to manufacturers is credibility. Manufacturers will be more trusted when trying to accumulate clients as their use of Latis services will stress their commitment to security. As such, manufacturers will be able to retain existing contracts while also maintaining the loyalty of their current clients. [31]

[31] Business Case - Diya, Isabella, Somya, Priya

Risk & Issues



Articulation and Rationale of Risks:

1. Government policy change: since many manufacturers that need cyber security have government contracts, the government imposing a requirement for no third party data transfers would make our solution concept obsolete. There is no discussion of this happening in the government currently making it less likely. Since it wouldn't close the entire market for us it isn't severely impactful.
2. Downtime during updates: since our solution concept is to minimize downtime during updates, not eliminate it, it is likely that downtime will happen to some degree. The impact is low because we plan on minimizing it by a significant margin, so remaining downtime shouldn't be as costly.
3. Copyright issues: if someone tries to utilize our idea, process, or logo as their own. The likelihood of this happening is rather low, since we would take necessary precautions to secure our ideas. But, if for some reason it does happen, the impact would be significant because we would have to spend resources guarding our owned information.
4. Inconsistencies with filepoint and Hadera: FilePoint and Hadera face occasional downtime. This is rare, so not very likely, but if it were to happen it would impact a significant portion of our solution concept.
5. Technical problems with TEEs: TEEs are created by a third party vendor, so if we install different TEEs into customers and they have technical issues, that would be a big risk. The likelihood of

that happening isn't extremely probable but could definitely happen if we choose the wrong vendor. The impact of this is higher because it would cost our company more money to replace and lose trust of customers.

6. Data breaches: Data breaches aren't likely to happen once our system is instituted, as its entire purpose is to increase safety. But, if our system did leave potential for data breaches that we hadn't considered, it would cost us customers. This would severely impact us.
7. Lost connections: If the internet goes down at our customer's facility, our semi-over the air system will fail. If connectivity between devices or the OEM and the manufacturer fails, our system fails too. This is unlikely as it would require environmental intervention, but if it were to occur it would be extremely significant to the functionality of our solution.
8. Supply chain shortages: Some of our solution concept involves adding physical components to existing systems. If these components suddenly become less producible, we struggle to implement a complete solution concept. Once again, it is not very likely, but since it is entirely environmental it would have a great impact.
9. Funding risk: since our solution concept requires testing and primary implementation this would create a cost prior to revenue streams. If we can't get primary funding, we would face an inability to create and implement our solution concept. While it isn't unlikely, it would severely impact the possibility of the creation of our solution.
10. Reluctance to adopt: if the market doesn't buy into our product, our solution concept becomes obsolete. Since the market does have a lot of high securities in place, and high security contracts, it is likely that resistance to adoption will occur. This would severely impact our project.

Risk Mitigation Plan:

Downtime during updates would be mitigated through data collection and analysis, and then optimization of slowing processes. Government Policy Changes would require reassessment of legality in our methods of security, ensuring that we align with current policies. Technical Problems with TEEs would require significant testing to see where the error is occurring, then either replace the brand used or determine the user error. Copyright Issues risks would be mitigated by preemptively acquiring copyrights for all of our designs and such. Reluctance to Adoption would be mitigated by a thorough examination of the rationale behind consumers reluctance. We would then change or improve upon our desirability attributes to account for the necessary changes. Funding Risk would be mitigated through reasonable planning and obtaining grants/loans if necessary. Supply chain shortages can be mitigated through ensuring we have backup suppliers for all our tools. Data Breaches would require a reassessment of our current system, with a focus on the weakness that caused the breach. We would then implement systematic changes to thwart that breach from recurring and implement them in subscribing customer's systems. Lost Connections would be addressed through realigning customer systems with the information transferring portion, and troubleshooting to determine which part or command caused the connection loss. We would then send a system update to ensure that a similar connective issue doesn't reoccur. In case there are inconsistencies with FilePoint and Hedera, we have a mitigation plan to have backup networks such as Ethereum, Solana, and Arweave.[32]

[32] Risk and Issues - Priya and Isabella

Trade Study

FOM/Solutions	Weight	Air-Gapped	Cloud	DLT
Speed (Quantitative)	0.1	1(0.1) = 0.1	9(0.1) = 0.9	6(0.1) = 0.6
Simple Installment (Quantitative)	0.15	6(0.15) = 0.9	9(0.15) = 1.35	3(0.15) = 0.45
User Usability (Qualitative)	0.15	3(0.15) = 0.45	8(0.15) = 1.2	8(0.15) = 1.2
Inherent Trust (Valued Feature)	0.35	1(0.35) = 0.35	6(0.35) = 2.1	9(0.35) = 3.15
Permission Access (Valued Feature)	0.25	1(0.25) = 0.25	7(0.25) = 1.75	9(0.25) = 2.25
Total	1	2.05	7.3	7.65

[33]

Rationale For Weight and Scoring:

For our Weighted Pugh Matrix, we utilized a scoring range from 1-9, 1 being lowest score and 9 being highest. Since air-gapped systems are the most prevalent today out of our three alternatives, we have much more research supporting our scoring. We will use air-gapped systems as our baseline in this Matrix.

We choose to look at three different solutions for our problem space due to their current prevalence in industry. The first was air-gapped updates which is heavily used in DoD contracting due to strict requirements set forth by the government. The second solution was cloud updates heavily used by large manufacturing companies due to the ease of use and limited amount of resources required to implement. The last being decentralized ledger technology which is mainly in the R&D phase and hasn't been implemented in the industry at this moment in time at a large scale.

For the features of merit we decided to look at five specific features: speed, simple installment, user usability, inherent trust, and permission access. The first factor is speed since it is important for updates to be implemented in a timely manner to reduce down times. This feature is important, but a small sacrifice in time is not a significant factor in comparison to the damage of security attacks. The criteria for speed is remaining under the 45min-1hr current industry standard. The second feature was simple installment, this is vital since easier installation allows for a lower adoption barrier. The criteria for installment would be the technical level that would be required by the installer. Even though this feature

[33] Trade Study Table - Isabella

is important, it only really plays a role in the beginning which in the long run has near negligible effects. The third feature is user usability, a technology may be amazing but if an user is unable to understand and use the interface the technology is useless. Even though a technology may be hard to use at first, most technology can be taught to help the user get accustomed which is why this weight is low. The fourth feature is inherent trust. This feature is weighted far higher than the other features due to the fact that this is the aspect that matters the most when ensuring updates are secure. The criteria for usability is how likely the average non-engineer could interpret the interface. The last feature is permission access, this part ensures that only the authorized user has access and is vital since it ensures that nobody can manipulate the update. This update is weighted heavily as well due to its importance in the overall security of the system. The criteria for this is from previous implementations of the potential solutions in industry.

For the air-gapped solution we gave it a rating of 1, 6, 3, 1, 1, since it requires a person to manually install the update, there are no real security protocols stopping anyone from installing an update/accessing the update, the protocol is typically proprietary for each machine, but overall the process is fairly simple to get up and running.

For the cloud solution we gave a rating of 9, 9, 8, 6, 7, since speed, installation, and usability are excellent. This solution is very simple and intuitive, however, when it comes to security this solution starts to lag a bit since it is vulnerable to single attacks that can take down the entire centralized network. We see a much more efficient approach in utilizing Cloud over air-gapping.

For the final solution, DLTs, we gave a rating of 6, 3, 8, 9, 9, since the speed is slower and installment would require more effort. However, similar to a cloud solution a user interface can be built to simplify the process but the system is built upon trust due to the decentralized nature which provides it with significantly higher ratings for the security criteria. This is a comparatively more secure approach than our baseline, while still maintaining more impactful speed.[34]

Sustainability

Sustainability is another main aspect to consider in this report. Our team needs to ensure that there are features aligned with the problem space and relevant to the solution concept. For starters, we know that it is desirable for our solution concept to be technologically advanced. As the world constantly changes and moves towards industry 4.0, our solution needs to stay up-to-date. To maintain sustainability we will create contractual updates where as the customer updates, our securities match those advancements.

Source	Category	Attribute to Consider	Comment/Artifact
Feasibility and Desirability	F(1) Stakeholder(D4)	Since several of our potential users manufacture parts for government contracts, our solution concept will be usable within government regulations.	Business Case/Systems Diagram /Functional and Physical decomposition
Feasibility and	F(2), Equipment	Our solution must remain competitive as the	Performance to or

[34] Trade Study - Soham & Isabella, Priya, Somya, Diya

Desirability and Viability	Suppliers(D2), Derived (D4), (V1,2,3).	industry develops towards 4.0, so we will create repeat reassessments of the market and implement innovations as they arise.	establishment of industry standards
Feasibility	F(3)	Manufacturers will always exist.	Infrastructure needed for approach to operate
Feasibility and Viability	(F1),(F2),(F3),(V 1,2,3,4)	A future application of our solution concept could be on power grids, creating a sustainable new market and providing a positive social impact. Our solution concept offers improved resilience, reliability, and cost-efficiency. This could be beneficial to public systems like power grids and train systems, creating securities to avoid downtime and attacks. This is a significant societal benefit.	Can your approach scale to other applications
Feasibility and Viability	(F4) (F5), (Vall)	People will choose Latis because they will save money using Latis as a preventative measure to cyberattacks.	Why will people choose your approach

Social Impact or Adoption

A future application of our solution concept could be on power grids, creating a sustainable new market and providing a positive social impact. Our solution concept offers improved resilience, reliability, and cost-efficiency. This could be beneficial to public systems like power grids and train systems, creating securities to avoid downtime and attacks. This is a significant societal benefit.

To meet the desirability requirement of user usability, we will ensure that our solution provides clear concise functions, with ability to adjust accessibility for different user needs. [35]

Economic Endurance

To maintain the cost efficiency desired of our solution, we will constantly reassess industry advancements to see if more cost efficient technologies arise. In doing this, we will update our systems to stay competitive with lowering costs and technology advances.

Environmental Endurance

Our solution concept doesn't focus on lessening businesses environmental impact, as the entire concept is based in intangible spaces. Our business model will shift towards more environmentally friendly suppliers in hopes to lesson our supply footprint within five years. The possibility of companies utilizing less energy during update times due to our reduced speeds, also helping the environment.

[35] Sustainability - Isabella, Priya, Somya, Diya

Sustainable Constraints and Barriers

Currently, political infrastructure has no barriers or constraints against cyber security efforts. But, as this industry develops we may see more laws pertaining to what you can and can't do with security. Another legal concern is that our primary market is heavily involved with government contracts. These government contracts are extremely confidential and require maximum security. If the government doesn't allow a third party to secure their contracts, our market shrinks significantly. Infrastructure won't be an issue as long as more manufacturers attempt to move towards industry 4.0.

Informative Sequence

Throughout our Four Set Analysis, we determined attributes from Feasibility, Viability, and Sustainability from the original desirability needs. As we referred back, we ultimately added more desirability requirements. This process of informative sequence is visualized in the Feasibility and Viability tables, with the cell referencing referring back to the specific desirability attributes that they respond to. As for sustainability, proof of informative sequence logic was shown as each attribute was in effort to solve or support desirability attributes aforementioned.

For a specific example, the need to minimize downtime and security costs is a key desirability attribute. This informs feasibility because our team opts to choose cheaper materials and methods that already exist in our solution concept. This also impacts viability because the ability to mitigate lost costs directly informs this set. Finally, to ensure that cost reduction is continued as the world updates and technology advances, contractual verification that we will reassess solution choices as new technology emerges informs sustainability. [36]

Causal Links

A causal link that must be considered exists between the Sustainability and Viability sets. In providing services to manufacturers like MOOG, upon whom the Business Case has been structured, it is important to note that government contracts contribute to a great deal of company projects. With regards to sustainability, in taking on government contracts, there is a degree and type of security guidelines that are required by the government agencies facilitating this arrangement. For example, MOOG is regularly awarded contracts of over $12 Million for the production of Aircraft Parts and Auxiliary Equipment by the Department of Defence. It is important to consider, for this reason, how the enabling technologies in our Feasibility set, and our overall design proposition will align with the requirements and regulatory policy of the end user (MOOG), transferred to it by its client, the US Government. This causal link directly impacts the target market scalability of our solution concept.

Desirability criteria is to be cost effective; however, third party contracts and licenses drive costs up. We need to reassess the extent to how much of a cost barrier we have and stakeholder leniency on minimized cost requirements.

Another link that exists is between the Feasibility and Viability sets. Although there are currently feasible options for storing and sending updates through cloud, it is important to consider how our solution

[36] Informative Sequence - Isabella

concept adopts a decentralized ledger technology to access a unique end user. In terms of feasibility, there are already pre existing solutions to our problem space. However, when linked to viability and given our scope, Latis is working with an untapped manufacturing sphere running mainly on legacy systems. Our solutions concept allows for ease of secured updates to our customer base.[37]

Solution Concept

Enabling Technology

For our solution concept, we are utilizing a decentralized ledger technology as our enabling technology to assist with streamlined updates for large-scale actuators in the manufacturing sector. Below is a concept of operation to help explain how each of the technologies described above interacts with each other in our solution concept from a high-level perspective.[38]

Solution Concept Rationale

Our solutions provides value through the security it is able to offer. As discussed in earlier sections, our competitors in the market utilize systems that are built on a centralized system inherently making them insecure due to the vulnerability of a single attack on the company that makes the entire network insecure.

Our solution offers a way around this through DLTs. This approach allows for a verification process requiring multiple entities and ensures no single point of failure is possible. Additionally, our solution allows for an update to be transferred directly from the OEM provider to the manufacturer without any interference from other actors. Specifically, it ensures that the integrity of the file sent from the OEM is upheld all the way to the machine itself, while still providing the manufacturer control over the process. This is unique compared to systems where the OEM directly sends updates since our system still allows for the manufacturer to make decisions on when and which updates to install.

Another important aspect is how the data is transferred through a singular portal. Our system allows for multiple 3rd Party OEM's to transfer data without interference from one another. This helps to simplify the process on the manufacturer side since it no longer requires proprietary portals for each manufacturer bringing about the idea of interoperability. [39]

Prototype Approach

The current status of the prototype design is here: https://github.com/LATISNetwork

As shown in the high level system design diagram, there are three main parts of the system: the OEM, the Middleman, and the Manufacturer. Each of these sections is necessary to the overall security and human-in-the-loop control and trust of the system; however, it has a repetitive stack that can be designed through full-stack development and testing.

[37] Causal Links - Priya, Diya, and Isabella
[38] Enabling Technology - Soham
[39] Solution Concept Rationale - Forrest

To start, the OEM and the Manufacturer both have a desktop application they can open and use for viewing and interacting with current updates, updates in queue, devices, and OEM/Manufacturer relations. The necessary features of these are that there is some sort of file accessibility for the OEM to upload files, some sort of compute power needed to encrypt the files, and intuitive reactive design for UX fundamentals.

Thus, we chose to use SvelteKit as the main web javascript framework to create the applications. The applications should be locally hosted, so that they are not susceptible to cloud based attacks on any of the operations. Additionally, they should be run on Electron browsers, since most modern browsers, such as Google Chrome and Mozilla, according to Mozilla Web standards, are not able to access local file paths. We use TailwindCSS for frontend design, Svelte for frontend frameworks, and SvelteKit for backend server APIs. Both clients are made with the same basic web stack.

The OEM and Manufacturer should only be able to access the application with proper authentication. Therefore, we introduce the hardware wallet, in the form of the Ledger Nano S at the moment. We are able to use the transport-webhid library to establish a USB connection with the Ledger as a peripheral. By interfacing with the Ledger, we can get data such as the public key of a wallet and sign transactions on chain with the Ledger's key. This allows for authentication and trusted access, similar to how current hardware based keys work. The whole implementation of Ledger access can be seen here: https://github.com/LATISNetwork/OEM-Client/tree/main/src/lib/components

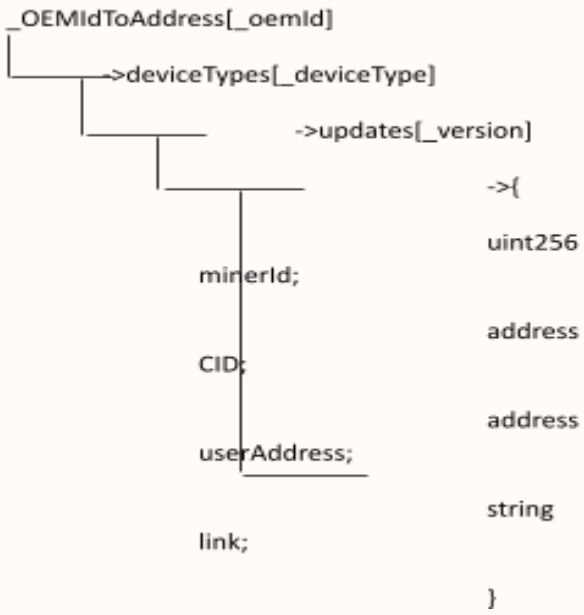
On the OEM side, we need a way to interface with the IPFS and Filecoin storage platform for decentralized storage of the data. Thus, we use Estuary (https://estuary.tech/), which has a node based client that we put in the backend api. This allows for fairly fast upload to IPFS and Filecoin through a REST API call, which then returns information such as the CID and link. The Filecoin miner is automatically found by the Estuary service, so we do not need to worry about that.

However, everything on IPFS is inherently public unless we set up a private cluster of miners; as long as someone knows the CID, they could access the files. In the optimal design, there would be a secret network of nodes and miners for storage, analogous to decentralized Amazon S3 storage. Therefore, we have E2E encryption for the files. We can use AES-256-CBC for encryption. We have a constant key that all clients know but a randomized salt, and we can encrypt the file to become a .enc file prior to sending it through Estuary. This is benchmarked at around 57 seconds for a 403MB file on a Raspberry Pi 3b, which shows its feasibility for speed. This service is called by an API call prior to the Estuary API call.

How can we send this information to the Manufacturer in a trusted and secure way? We can use the Hedera hashgraph system to send transactions, which are cheap and fast, to the Manufacturer to sync data. The smart contracts currently are shown here: https://github.com/LATISNetwork/SmartContracts.

There are three main contracts at the moment: the OEM, the Middleman, and the Manufacturer. The structure of each smart contract is similar. To manage the rules and permissions of the contract, we use the OpenZepplin protocol for Solidity, where we assign roles to certain addresses. This enables secure access and execution of contract methods, voiding possibilities of outside agents abusing or attacking the contracts. The OEM contract is one the OEM can directly sign transactions to with their Hedera

wallet. With proper authorization, they can push updates that provide information about the file checksum, links, CID, version, and target devices. This is pushed to a MiddleMan contract (accessible only by other contracts), which then acts like a stores on chain for information. The information of updates is stored in a tree structure of:



The MiddleMan contract also has proper ADMIN methods to add and proper removal and getter methods.

Once the OEM has successfully uploaded the file to the FILE network, we need a cost efficient way of notifying the Manufacturer that there is a new update available. We can use Firebase (or something similar) for simple notifications; doing this through chain is too expensive and not necessary, as long as we do not send any confidential information.

On the Manufacturer side, given a notification, they can then call their contract to refresh and fetch new update information stored on the MiddleMan contract, which is then reactively shown on the client to send to the proper IoT devices. In an internal based network, which in the best case scenario be a locally hosted DAG/hashgraph based consensus framework like Hedera or Hyperledger with IoT devices as nodes, the Manufacturer sends the key information, such as the CID and update requirements, to the IoT device. With the private key that is fused into the TEE in the secure world of the ARM CPU (Trustzone technology), we can sign transactions with the contracts. For our implementation, we will be using OP-TEE or TF-A, both of which are protocols where we can have direct access to the TEE for prototyping and development uses. OP-TEE is open source, while TF-A is one of ARM's main trusted firmware development systems.

The IoT device, given it is connected to the internet or has access to a miner on an internal network, can then download the proper file on edge. With the predetermined key and the new salt that is sent along

with the data, the IoT device can perform a decrypted AES-256-CBC decryption. On the Raspberry Pi 3b, it is benchmarked less than 1 minute for a 403MB file.

With the firmware update downloaded and ready on the peripheral IoT device, like an RTU or Raspberry Pi adjacent microcomputer, we can perform a PXE (Preboot eXecution Environment) boot on the main device computer. We are assuming the main device computer has an ARM or ARM adjacent architecture. In ARM based boot systems, there are three main bootloader levels from a cold boot. The first stage is BL1, which is fused into the ROM code of an ARM chip, and starts when the platform is physically turned on. This sets up SRAM for future levels and also starts the printf process for BL2. BL2 sets up RAM and initializes the initial architecture and initializes BL3. Here, the chip then initializes UART and USB; this is where our PXE boot can take place. The PXE boot updates either the whole OS or key files and registers that monitor firmware methods, such as PSCI (Power State Coordination Interface) and SCMI (System Control and Management Interface). All of this is performed in the secure world. In BL33, the kernel starts and can finally finish up the bootloader sequence. This is all through wire connection and edge bootloading, so there is no wireless based attack surface.

Overall, we have three main methods of security and trust for OTA updates. The first is E2E encryption from the OEM to the IoT device. The second is the DLTs we are using, which include Filecoin and Hedera as well as the hardware Ledger wallet. By using consensus, we can trust all parties involved in the transaction without restricting or opening third party (OEMs) access without continuous monitoring. We can also make sure there are no middle-man based attacks between either the OEM and the Manufacturer, and the Manufacturer to the IoT device. Finally, the trusted environment on the IoT device ensures that only it can confirm receiving an update or order that is directed to it, by signing transactions and update sequences through hardware-fused keys. [40]

Value Proposition

For manufacturing companies, who require a secure method to update large, computer-controlled actuators, a decentralized ledger network provides a secure and trusted streamlined pipeline for manufacturers to update third party OEM machinery.[41]

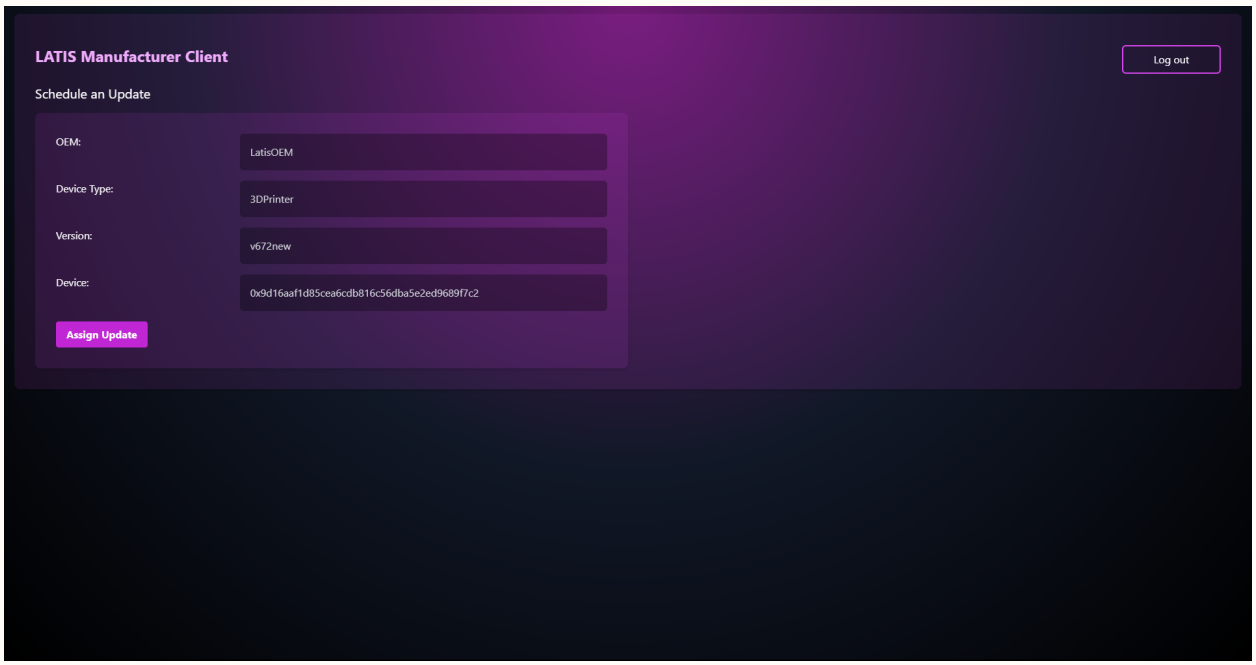
[40] Prototype Approach - Forrest
[41] Value Proposition - Soham

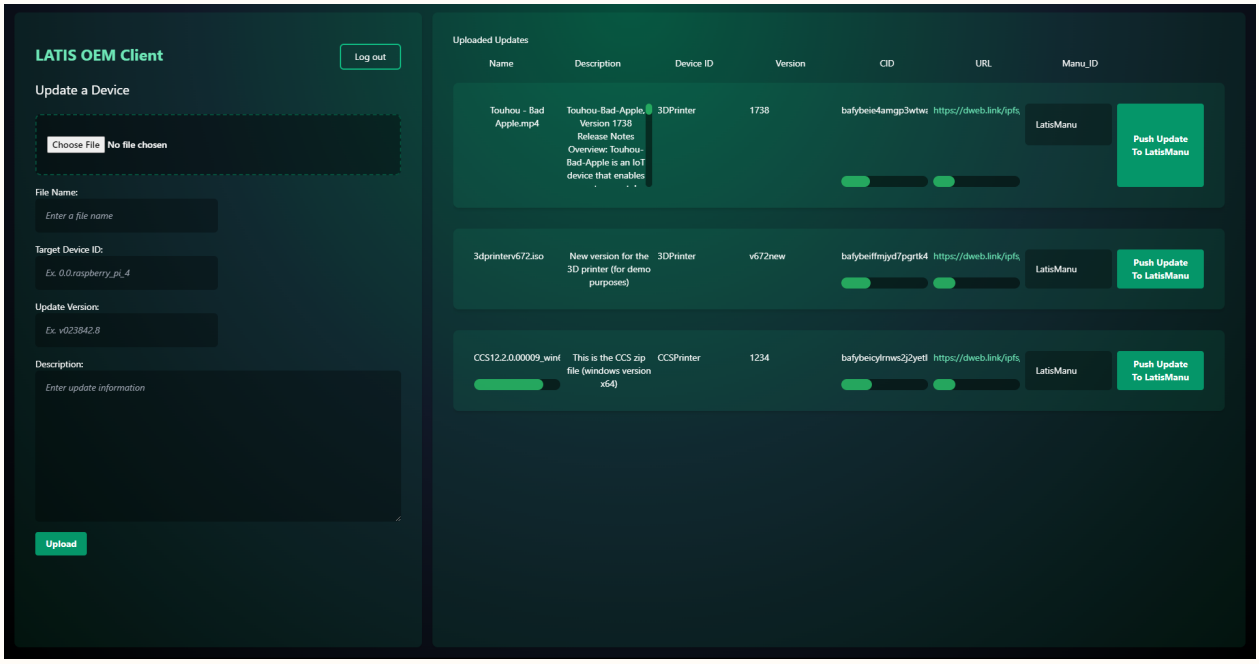
Prototype Artifacts

Expo Demonstration

The above photograph contains the entire prototype. The white computer represents the OEM client and the black computer represents the Manufacturer. The OEM client also has a physical Ledger S Nano Plus connected to it which acts as their keys to authenticate the user and enable access to the software. There is also a Raspberry Pi 3 B+ connected to the monitor. At the high level, the entire execution of the prototype is as follows: the user can log in on the OEM client side with the ledger, they can upload and push a “firmware” update for the raspberry pi, on the Manufacturer side, they can choose a device to

assign the received update to (in this case the raspberry pi), and finally, they can send an update to the pi and watch as the device reboots on the monitor after receiving the update.[42]





Prototype Deep Dive

There are three main smart contracts that run the backend authentication for updates; the OEM, the Middleman, and the Manufacturer contracts. The OEM and Manufacturer are accessed by their

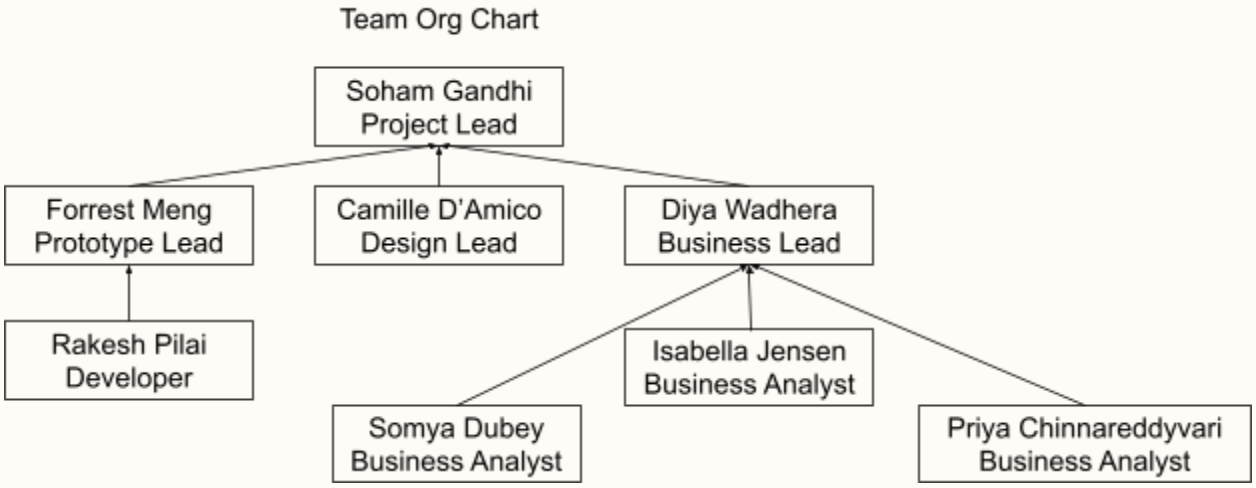
[42] Expo Demonstration & Pictures - Rakesh, Forrest

respective actors, while the middleman contract is only accessible by authorized smart contracts. We used Ledger Nano S Plus as hardware keys for the OEM and the Manufacturer to authorize access to the portals, as shown below. Once they log in using the Ledger and pass the OAuth screen, they are met with their respective portals (green is OEM, fuschia is Manufacturer).

The OEM portal includes a component to upload an update file to Filecoin. When all of the metadata and the file are filled in, the backend first performs a simple block cipher on the file and then stages it on IPFS to upload to Filecoin, using the Estuary node client. It returns its CID, the miners, and the URL to access the file. Additionally, the metadata is pushed to the OEM smart contract, which then updates the middleman contract with relevant information. When the OEM "pushes" an update to specific manufacturers, the middleman contract would send a notification, similar to an interrupt, to the Manufacturer, signaling that there is a new update.

On the manufacturer portal, they can then select all the basic information to facilitate the update, including the OEM, the type of device, the version of the update, and the device ID (MAC address or UUID). Assigning the update would notify the IoT device that there is an update waiting for it, and it would then pull the information from the middleman contract to pull the data off of Filecoin. Then, it would automatically decrypt and apply the update. In our scenario for the demo, we had a rpi 3b act as an RTU, pull an update, decrypt it, and PXEBoot another rpi 3b. Nowhere in this entire pipeline does the manufacturer or any other actor have access to the pushed update, aside from the IoT device. Additionally, everything that we could put on a DLT was put on a DLT and encrypted/secured (either ciphers or OpenZepplin), maximizing the overall security of the system.[43]

Team Structure



[43] Prototype Deep Dive - Forrest, Rakesh

Conclusion

Our current solution concept addresses a need for security, efficiency, and connectedness within the manufacturing industry. As they adopt and develop into Industry 4.0, a closed environment is inefficient, and manual updates create significant downtime costs.

Our solution is an interconnected system that supports and secures over-the-air updates. Our specific market is mid-sized manufacturers who have outdated systems and equipment. Through the utilization of several checking procedures (enumerated inside of Solution Concept sections), we expect a significant decrease in overall downtime losses and potential cyber attack. Our solution meets each of our stakeholder's desirability needs, and forms a cohesive response in a feasible, viable, and sustainable manner.

As we further develop our solution concept, a complex, meaningful prototype will materialize and we will begin thorough testing. Our next steps include further interviews with stakeholders to better understand how our solution can help them or how we can shape our solution to better meet the needs of the market. In the near future we plan to test our solution at the learning factory, getting a chance to expose our solution to a simulated real world situation where stress tests can be implemented to demonstrate the security our solution has to offer. [44]

[44] Conclusion - Isabella

References

(n.d.). Trusted Execution Environment, - TrustZone and Mobile Security. Retrieved April 2, 2023, from https://owasp.org/www-pdf-archive/OWASP_Security_Tapas_-_TrustZone,_TEE_and_Mobile_Security_final.pdf

(n.d.). SmartAxiom: Home. Retrieved April 2, 2023, from https://www.smartaxiom.com/

(2019, September 9). CS Info Template. Retrieved April 2, 2023, from https://media.defense.gov/2019/Sep/09/2002180319/-1/-1/0/Update%20and%20Upgrade%20Software%20Immediately.docx%20-%20Copy.pdf

Ahmadi-Assalemi, G., & Al-Khateeb, H. (2022). Blockchain technologies in the design of Industrial Control Systems for Smart Cities. *IEEE Blockchain Technical Brief*. https://blockchain.ieee.org/images/files/pdf/techbriefs-2022-q2/blockchain-technologies-in-the-design-of-industrial-control-systems-for-smart-cities.pdf

Anderson, S. (2023, February 9). *What Is Proof of Work (PoW) in Blockchain? - Bitcoin*. Investopedia. Retrieved April 2, 2023, from https://www.investopedia.com/terms/p/proof-work.asp

Arghire, I., Solomon, M., Wilson, M., Antova, G., Naraine, R., Kovacs, E., & Townsend, K. (2016, November 18). *Over-the-Air Update Mechanism Exposes Millions of Android Devices*. SecurityWeek. Retrieved April 2, 2023, from https://www.securityweek.com/over-air-update-mechanism-exposes-millions-android-devices/

Baird, L. (n.d.). *Hedera Hashgraph vs Blockchain | Comparison*. LeewayHertz. Retrieved April 2, 2023, from https://www.leewayhertz.com/hashgraph-vs-blockchain/

BlackEnergy APT Attacks | What is BlackEnergy? (n.d.). Kaspersky. Retrieved April 2, 2023, from https://usa.kaspersky.com/resource-center/threats/blackenergy

Boeing. (n.d.). SIMBA Chain. Retrieved April 2, 2023, from https://simbachain.com/case-study/boeing/

CMMC 2.0 Certification Costs. (2022, May 2). Ignyte Assurance Platform. Retrieved April 2, 2023, from https://www.ignyteplatform.com/cmmc-2-0-certification-costs/

A complete guide to NIST cybersecurity framework. (n.d.). Hackcontrol. Retrieved April 2, 2023, from https://hackcontrol.org/blog/nist-cybersecurity-framework/

CrashOverride Malware. (2017, June 12). CISA. Retrieved April 2, 2023, from https://www.cisa.gov/uscert/ncas/alerts/TA17-163A

Cyber-Attacks on Industrial Assets Cost Firms Millions. (2022, June 2). Trend Micro | Newsroom. Retrieved April 2, 2023, from https://newsroom.trendmicro.com/2022-06-02-Cyber-Attacks-on-Industrial-Assets-Cost-Firms-Millions

Cyber risk in advanced manufacturing. (n.d.). Deloitte. Retrieved April 2, 2023, from https://www2.deloitte.com/content/dam/Deloitte/us/Documents/manufacturing/us-manu-cyber-risk-in-advanced-manufacturing.pdf

Cybersecurity of Firmware Updates. (n.d.). NHTSA. Retrieved April 2, 2023, from https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/cybersecurity_of_firmware_updates_oct2020.pdf

Difference Between Blockchain and Hashgraph. (2022, August 16). GeeksforGeeks. Retrieved April 2, 2023, from https://www.geeksforgeeks.org/difference-between-blockchain-and-hashgraph/

The Difference Between HTTPS and SFTP. (2022, February 24). GoAnywhere MFT. Retrieved April 2, 2023, from https://www.goanywhere.com/blog/https-vs-sftp-the-key-differences

Empey, C., & Latto, N. (2020, April 8). *VPN Meaning: What Is a VPN & What Does It Do?* Avast. Retrieved April 2, 2023, from https://www.avast.com/c-what-is-a-vpn

Froese, B. (2022, June 23). *Blockchain Gets Serious: Industrial Use Cases of Distributed Ledger Technology*. Detecon International. Retrieved April 4, 2023, from https://www.detecon.com/en/journal/blockchain-gets-serious-industrial-use-cases-distributed-ledger-technology

Fruhlinger, J. (2022, August 31). *Stuxnet explained: The first known cyberweapon*. CSO Online. Retrieved April 2, 2023, from https://www.csoonline.com/article/3218104/stuxnet-explained-the-first-known-cyberweapon.html

HBAR (ℏ). (n.d.). Hedera. Retrieved April 2, 2023, from https://hedera.com/hbar

Home. (n.d.). YouTube. Retrieved April 2, 2023, from https://learn-cloudsecurity.cisco.com/umbrella-library/2021-cyber-security-threat-trends-phishing-crypto-top-the-list#page=1

IBM Report: Manufacturing Felt Brunt of Cyberattacks in 2021 as Supply Chain Woes Grew. (2022, February 23). IBM Newsroom. Retrieved April 2, 2023, from https://newsroom.ibm.com/2022-02-23-IBM-Report-Manufacturing-Felt-Brunt-of-Cyberattacks-in-2021-as-Supply-Chain-Woes-Grew

IBM Security X-Force Threat Intelligence Index 2023. (n.d.). IBM. Retrieved April 2, 2023, from https://www.ibm.com/reports/threat-intelligence/

Industrial IoT. (n.d.). IOTA. Retrieved April 2, 2023, from https://www.iota.org/solutions/industrial-iot

Internet, TCP/IP, and HTTP concepts. (2020, December 18). IBM. Retrieved April 2, 2023, from https://www.ibm.com/docs/en/cics-ts/5.2?topic=web-internet-tcpip-http-concepts

Introduction to Trusted Execution Environment: ARM's TrustZone. (2018, June 19). Quarkslab's blog. Retrieved April 2, 2023, from https://blog.quarkslab.com/introduction-to-trusted-execution-environment-arms-trustzone.html

IOTA Enters Next Phase of EU Blockchain PCP. (2023, January 27). IOTA Foundation Blog. Retrieved April 2, 2023, from https://blog.iota.org/iota-eu-blockchain-pcp/

Kumar, B. (2022, November 11). *What Is Data Encryption: Algorithms, Methods and Techniques [2022 Edition]*. Simplilearn. Retrieved April 2, 2023, from https://www.simplilearn.com/data-encryption-methods-article

Long-term crypto threat: quantum computers hacking bitcoin wallets. (2021, June 10). CNBC. Retrieved April 2, 2023, from https://www.cnbc.com/2021/06/10/long-term-crypto-threat-quantum-computers-hacking-bitcoin-wallets.html

Maciuca, D. (n.d.). *Over-the-Air Software Updates*. Ford Corporate. Retrieved April 2, 2023, from https://corporate.ford.com/articles/products/over-the-air-software-updates.html

McAfee Night Dragon Report (Update A). (2018, September 6). CISA. Retrieved April 2, 2023, from https://www.cisa.gov/uscert/ics/advisories/ICSA-11-041-01A

Network Protocols & How They Can Benefit Your Business. (2022, August 8). CDW. Retrieved April 2, 2023, from https://www.cdw.com/content/cdw/en/articles/networking/types-of-network-protocols.html

New survey reveals $2 trillion market opportunity for cybersecurity technology and service providers. (2022, October 27). McKinsey. Retrieved April 2, 2023, from https://www.mckinsey.com/capabilities/risk-and-resilience/our-insights/cybersecurity/new-survey-reveals-2-trillion-dollar-market-opportunity-for-cybersecurity-technology-and-service-providers

Nguyen, D. L., Bröring, A., & Pizzol, M. (2022). Analysis of distributed ledger technologies for industrial manufacturing. *Scientific Reports*, *12*. https://doi.org/10.1038/s41598-022-22612-3

NVIDIA DRIVE OS SDK Development Guide. (2019, November 6). NVIDIA Docs. Retrieved April 2, 2023, from https://docs.nvidia.com/drive/drive_os_5.1.6.1L/nvvib_docs/index.html#page/DRIVE_OS_Linux_SDK_Development_Guide/Windows%20Systems/security_concepts.html

Odum, F. (2022, November 16). *Report: Unlocking The Cybersecurity Landscape | A Contrary Research Deep Dive*. Contrary Research. Retrieved April 2, 2023, from https://research.contrary.com/reports/unlocking-the-cybersecurity-landscape

Over-the-air software updates (OTA). (n.d.). Polestar. Retrieved April 2, 2023, from https://www.polestar.com/us/manual/polestar-2/2021/article/Over_the_air-software-updates-(OTA)/

Production Monitoring. (2022, November 28). Zerynth. Retrieved April 2, 2023, from https://zerynth.com/solutions/industrial-iot/industrial-process-monitoring-and-optimization/

Ranger, S. (2020, February 3). *What is the IoT? Everything you need to know about the Internet of Things right now*. ZDNET. Retrieved April 2, 2023, from https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know-about-the-iot-right-now/

Regenscheid, A. (2018, May 2). *Platform Firmware Resiliency Guidelines*. NIST Technical Series Publications. Retrieved April 2, 2023, from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-193.pdf

SNMP Ports & Protocol - What is it? (n.d.). ThousandEyes. Retrieved April 2, 2023, from https://www.thousandeyes.com/learning/techtorials/snmp-simple-network-management-protocol

Software Updates. (n.d.). Tesla. Retrieved April 2, 2023, from https://www.tesla.com/ownersmanual/model3/en_us/GUID-A5A60CB3-7659-4B08-B2FD-AFD12C2D6EE1.html

Software Updates. (n.d.). Tesla. Retrieved April 2, 2023, from http://www.tesla.com/ownersmanual/model3/en_us/GUID-A5A60CB3-7659-4B08-B2FD-AFD12C2D6EE1.html

Vulnerable SDK components lead to supply chain risks in IoT and OT environments. (2022, November 22). Microsoft. Retrieved April 2, 2023, from https://www.microsoft.com/en-us/security/blog/2022/11/22/vulnerable-sdk-components-lead-to-supply-chain-risks-in-iot-and-ot-environments/

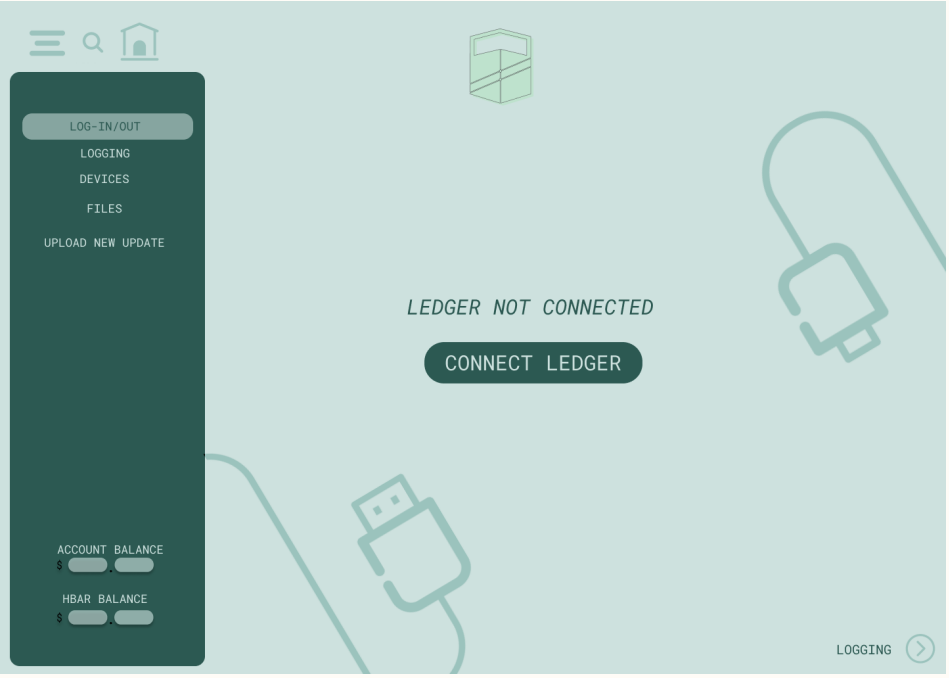
Walmart Case Study – Hyperledger Foundation. (n.d.). Hyperledger. Retrieved April 2, 2023, from https://www.hyperledger.org/learn/publications/walmart-case-study

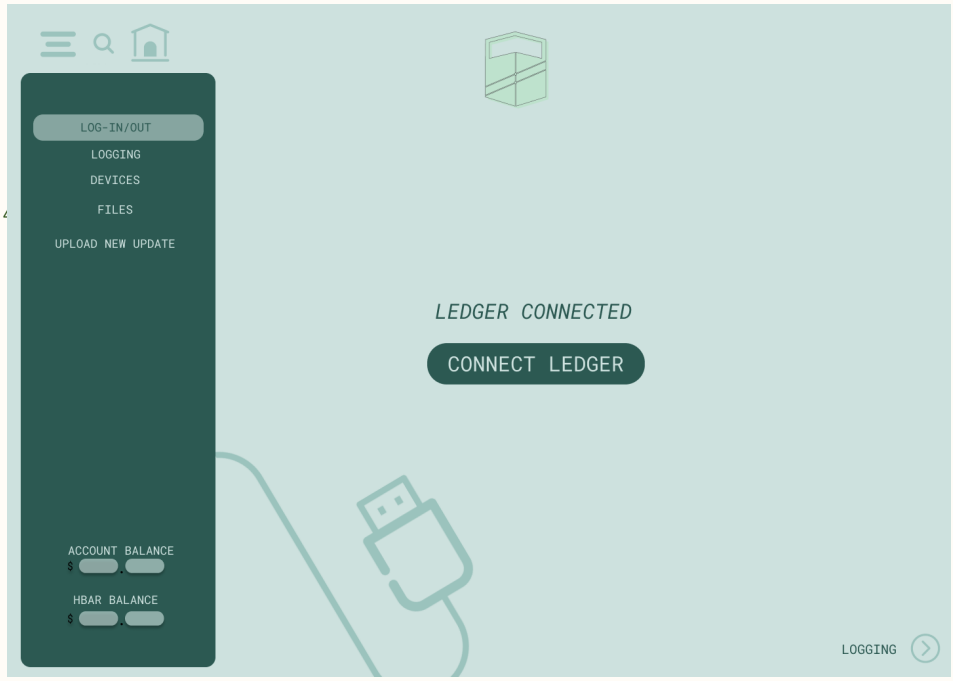
What is a software-defined perimeter? | SDP vs. VPN. (n.d.). Cloudflare. Retrieved April 2, 2023, from https://www.cloudflare.com/learning/access-management/software-defined-perimeter/

What is ICMP? | Internet Control Message Protocol. (n.d.). Cloudflare. Retrieved April 2, 2023, from https://www.cloudflare.com/learning/ddos/glossary/internet-control-message-protocol-icmp/

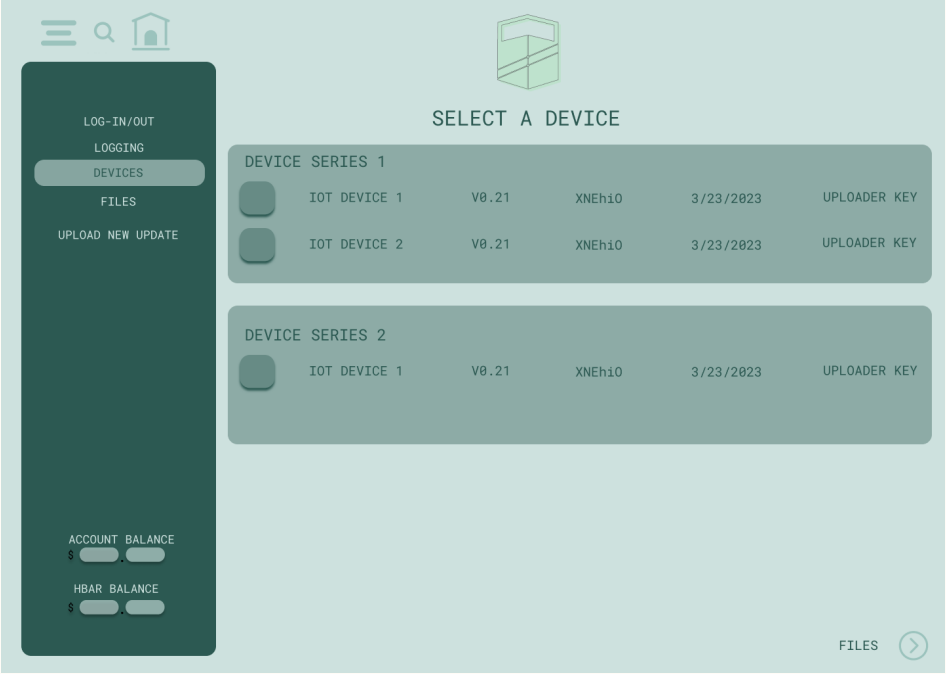
What Is Network Segmentation? (n.d.). Cisco. Retrieved April 2, 2023, from https://www.cisco.com/c/en/us/products/security/what-is-network-segmentation.html

Appendix

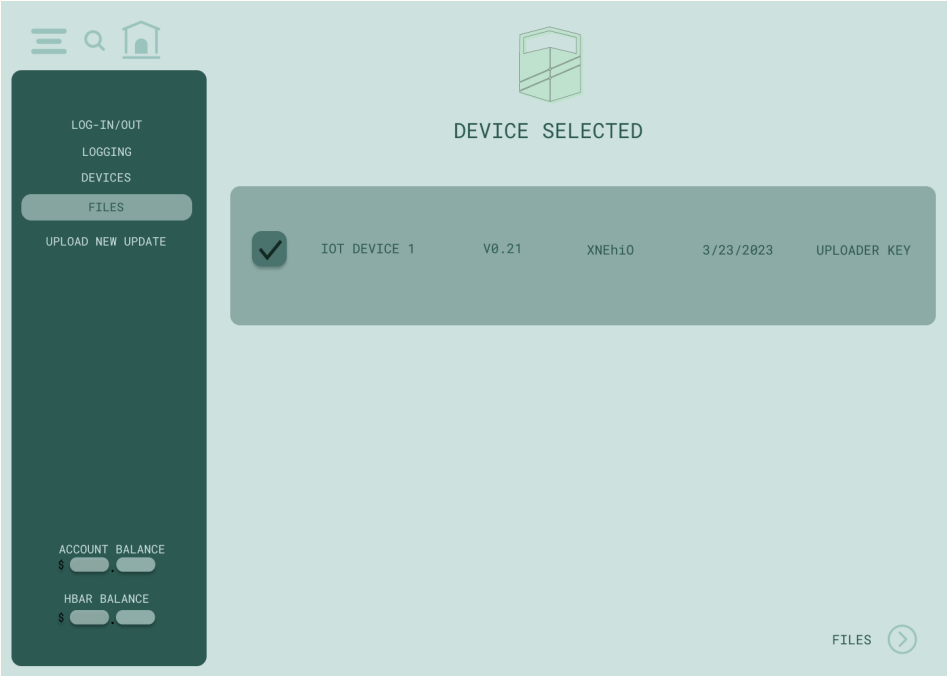




[45] Communications Design of UIUX Model - Camille D'Amico



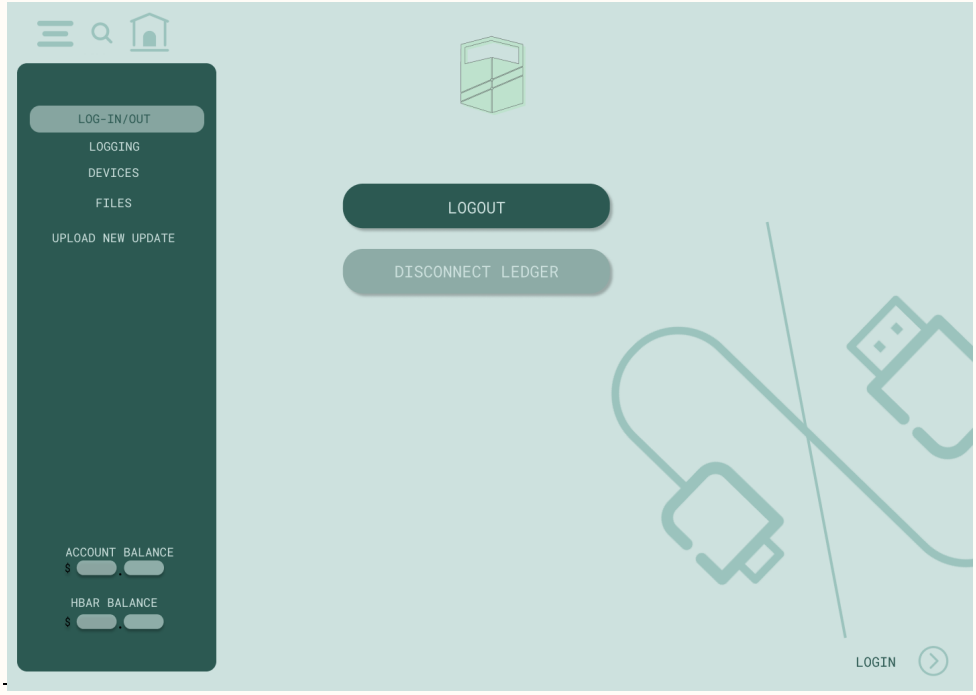
[46] Communications Design of UIUX Model - Camille D'Amico



[47] Communications Design of UIUX Model - Camille D'Amico


LOG-IN/OUT
LOGGING
DEVICES
FILES
UPLOAD NEW UPDATE
ACCOUNT BALANCE
$
HBAR BALANCE
$
NAME
STATUS
LOGGING LEVEL
NAME OF FILE
DEFAULT
NAME OF FILE
DEFAULT
NAME OF FILE
DEFAULT
NAME OF FILE
DEFAULT
HOME

48


LOG-IN/OUT
LOGGING
DEVICES
FILES
UPLOAD NEW UPDATE
ACCOUNT BALANCE
$
HBAR BALANCE
$
LOGOUT
DISCONNECT LEDGER
LOGIN

-

[49] Communication Design of UIUX Model - Camille D'Amico