Title

Secure Cloud-based Provisioning and Managing the Complete Life Cycle of IoT Metals

Permalink

https://escholarship.org/uc/item/87d2254z

Authors

Moghaddassian, Morteza

Garcia-Luna-Aceves, J.J.

Publication Date

2024-10-29

Peer reviewed

Secure Cloud-based Provisioning and Managing the Complete Life Cycle of IoT Metals

Morteza Moghaddassian
Electrical and Computer Engineering Department
University of Toronto
Ontario, Canada (M5S 3G4)
Email: m.moghaddassian@utoronto.ca

J.J. Garcia-Luna-Aceves
Electrical and Computer Engineering Department
University of Toronto
Ontario, Canada (M5S 3G4)
Email: jj.garcialunaaceves@utoronto.ca

***Abstract*—TIPS (Toronto Infrastructure Provisioning System) is introduced as a new cloud-based IoT metal provisioning and life-cycle management system. TIPS splits the life cycle of IoT metals (devices) into three provisioning phases. In Phase 1, an IoT metal undergoes a tethered hardware provisioning process that moves the non configured metal hardware from its raw state to a configured state (i.e., having a full-fledged operating system). In Phase 2, TIPS provides the configured IoT metal with subsequent provisioning of the software packages and binaries that are specifically intended to configure the metal for a given IoT system task (e.g., data processing, sensing, and actuating). In Phase 3, TIPS continues to provide the now-functioning IoT metal with more software updates such as program upgrades and security patches to meet the metal's ongoing software needs.**

TIPS multi-phase provisioning model improves the security of IoT metal provisioning by authenticating and authorizing IoT metals before each provisioning phase, reducing the impacts of security threats like unauthorized OS installation and mass deployment of malicious software stacks on IoT metals. TIPS offers granular control over the IoT metal provisioning process that enhances the timeliness of IoT provisioning systems for reacting to changes in security requirements. TIPS multi-phase provisioning model only marginally increases the provisioning time and energy consumption of IoT metals during the provisioning process while offering many security benefits.

I. Introduction

Traditional approaches for provisioning and managing IoT metals (devices) life cycle involve careful and manual configuration of the metals hardware before final deployment in an IoT system with subsequent field visits made to the metals for maintenance and applying post configurations. These approaches are typically time-consuming, insecure, and inefficient at scale. Modern approaches aim to use safer and smarter methods that leverage unattended configuration mechanisms and secure automatic updating techniques to reduce manual work and operational expenses (OPEX).

The state of the art in unattended configuration of IoT metals primarily relies on cloud Over-the-Air (OTA) software provisioning services [1], [2] that allow preconfigured IoT metals (i.e., having an operating system) to download and install software updates (e.g., security patches, OS kernel upgrades, and user programs) from a remote update server using popular and secure Web and machine-to-machine messaging protocols like HTTPs and MQTT [3]. For unattended configuration of IoT metals in raw/bare hardware state (i.e., having no operating system), a separate and additional metal hardware provisioning system must be typically used that includes more fundamental hardware configuration and software installation steps [4].

As Fig. 1 illustrates, the metal hardware provisioning process begins by IoT metals running in raw/bare hardware state to preset booting from a local system partition to boot from the network using PXE network stack [5]. The Preboot Execution Environment (PXE) network stack available over Ethernet, creates a client-server execution environment that allows IoT metals to obtain an IP address and the address of a remote hardware provisioner using BOOTP/DHCP [6], [7] as shown in Fig. 1, Steps 1 to 3.

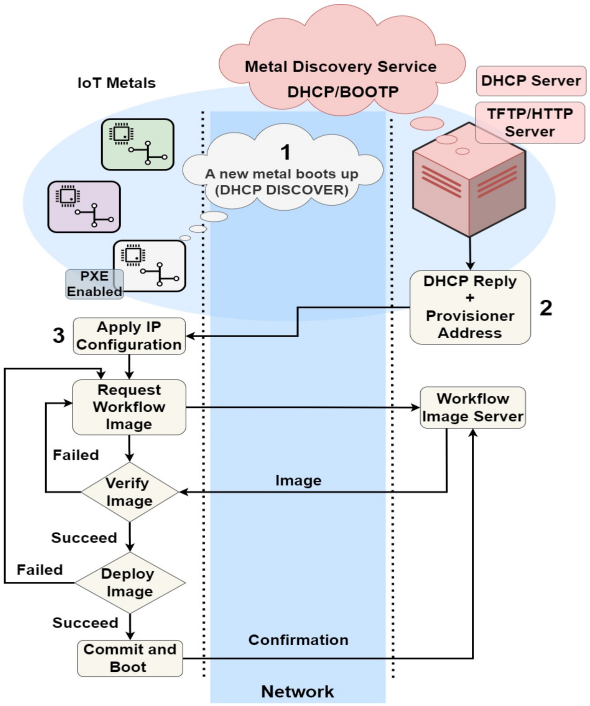


Fig. 1. Standard steps in IoT metal hardware provisioning process.

Once configured with an IP address, the PXE stack, in Stage 1 booting, downloads the Network Boot Program (NBP) to the IoT metals memory using TFTP [8], [9]. Next, NBP takes

IoT metals to Stage 2 booting wherein IoT metals can use the address of the remote hardware provisioner they obtained in Stage 1 to contact and download pre-defined workflow images over FTP or by using HTTP/HTTPs [4]. These workflow images are typically composed of an operating system, drivers, bins and developer libraries, and optionally user programs and packages that define the IoT metals function(s).

Today, IoT configuration and life-cycle management systems typically provide a combination of OTA and metal provisioning services to meet IoT metals software needs at different configuration stages. A key challenge, however, is maintaining the security of IoT metals provisioning to avoid installing malicious or unauthorized IoT metals software especially when provisioning IoT metals in raw/bare stage as such IoT metals do not have an operating system or bootable software stack to use for reliable authentication.

We propose Toronto Infrastructure Provisioning System (TIPS) to tackle the security challenges of IoT metals provisioning. TIPS is a cloud-based IoT metal provisioning and life cycle management system that provides a secure multi-phase IoT metal provisioning model. The goal in TIPS and its multi-phase metal hardware provisioning model is to provide increased control over the IoT metals provisioning process and to improve the security practices in cloud-based provisioning systems while delivering a complete life cycle (configuration) management of IoT metals.

The TIPS multi-phase provisioning model is composed of three phases. In Phase 1, non configured IoT metals undergo a tethered hardware provisioning process that moves the metals from raw/bare hardware state to a configured state (having a full-fledged operating system). In Phase 2, TIPS provides configured IoT metals from Phase 1 with additional provisioning of the user-specific packages and programs. This phase is intended to prepare configured IoT metals from Phase 1 to perform tasks like sensing and actuating that define their roles in the IoT systems. Finally, in Phase 3, TIPS continues to provide the now-functioning IoT metals from Phase 2 with additional support (e.g., security patches and program updates) to meet their ongoing software needs. Before each phase begins, TIPS ensures that only authenticated and authorized IoT metals can proceed.

TIPS uses a two-tier IoT metal discovery service, TIPS-D, which enables TIPS to register and assign globally unique identifiers (TIPS IDs) to discovered IoT metals and to authenticate and authorize these IoT metals before each phase. The device discovery service also registers multiple network interfaces of an IoT metal separately and ensures that malicious IoT metals cannot be provisioned by changing their environment or the IoT system they are currently deployed.

TIPS also introduces the use of Glaze images. Glaze images are self-executable binaries that enable the delivery of TIPS multi-phase IoT metal provisioning model. We show in section II that using Glaze images and TIPS multi-phase provisioning model can interestingly costs only a marginally longer time and higher energy consumption from IoT metals to become fully functional (i.e., passing Phase 1 and 2 provisioning) compared with a one-time provisioning approach that installs all software pieces on the IoT metals at once. Glazes and the TIPS multi-phase provisioning model can also improve the safety and security of IoT metals by preventing malicious software stack deployments on IoT infrastructures and enhancing the timeliness of cloud-based IoT provisioning systems to respond to changes in security requirements.

In terms of the portability of the provisioning process, TIPS Phase 1 provisioning requires IoT metals to be wired since PXE booting is only available over Ethernet [5]. However, TIPS can perform Phase 2 and Phase 3 of metal provisioning in wireless settings using Wi-Fi and OTA updating mechanisms. TIPS can also support multi-homing for the delivery of the Glaze binaries to IoT metals if an IoT metal have more than one registered network interfaces with TIPS-D. In the future, we hope that we can span the TIPS support of wireless technologies for OTA updating of IoT metals in Phase 2 and 3 to also include Bluetooth and LoRa wireless.

The rest of this paper is organized as follows. Section II presents the TIPS architecture that explains IoT metals discovery, registration, and multi-phase metal provisioning approach. Section II also provides the evaluation results of applying the TIPS multi-phase IoT metal provisioning model on Raspberry Pi 4 devices and explains the security benefits of TIPS multi-phase provisioning model. Section III provides a summary of existing work in the area of IoT metal provisioning and life cycle management, and Section IV presents our conclusions.

II. TIPS

The TIPS architecture is composed of three main components, as Fig. 2 illustrates:

TIPS Master (TIPS-M): is the main component of the TIPS architecture, and includes TIPS-D (TIPS IoT metal discovery and registration service), TIPS-U, and the TIPS Controller. TIPS-U is the TIPS IoT metal updater service responsible for the provisioning of registered IoT metals in Phases 1 to 3. The TIPS Controller performs management of users and IoT metals configuration, manages installation of Glaze binaries, and provides APIs for users to interact with the TIPS Master. The TIPS Master is designed cloud-ready and can be easily integrated with cloud computing frameworks. The TIPS Master software is also designed for containerized deployment on virtual machines and cloud physical servers, as shown in Fig. 2.

TIPS Gateway (TIPS-G): is an important on-premise component of the TIPS architecture. TIPS-G must be placed in the same network environment where IoT metals/devices are deployed, and plays three important roles. First, it works with TIPS-D to assist with IoT metal discovery and registration process. Second, it acts as the default gateway for IoT metals when interacting with the TIPS Master and external networks and provides minimal firewall support. Third, it functions as a NAT server for IoT metals within the same local network. TIPS-G is connected to TIPS Master via the Internet and can support wide ranging Internet connections including 5G and beyond, as shown in Fig. 2.

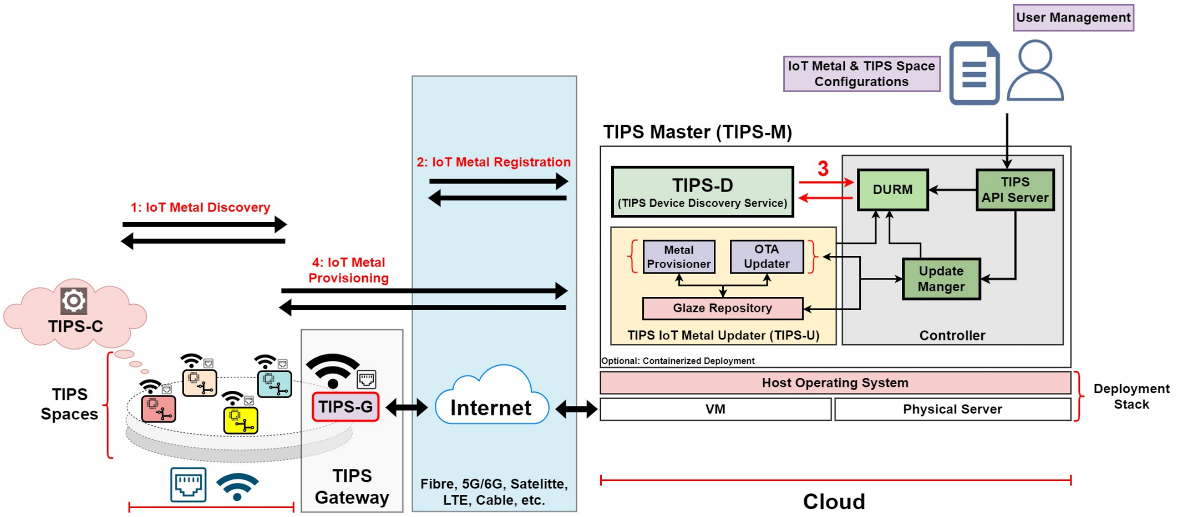


Fig. 2. A high-level view of the TIPS architecture.

TIPS Client (TIPS-C): is a client program that runs on IoT metals to receive, verify, and deploy software updates during Phase 2 and Phase 3 of TIPS multi-phase provisioning model. The TIPS Client will be installed on registered IoT metals during the Phase 1 of TIPS multi-phase provisioning model with the operating system installation.

A. IoT Metal Discovery and Registration

The provisioning process begins by IoT metals to first join TIPS. Prior to joining TIPS, users must create TIPS Metal Objects for their IoT metals/devices. A TIPS Metal Object (TMO) is an in-cloud object (i.e., a metal/device twin) maintained by the TIPS Master that holds information about an IoT metal configuration and its network interfaces, as shown in Fig. 3. Once created, each object is assigned a globally unique TIPS ID. Users must also provide the hardware addresses (e.g., MAC addresses) of their IoT metals during the process of Metal Object creation. If an IoT metal has multiple network interfaces, hardware addresses for all or a subset of the network interfaces that the metal can use for provisioning must be provided. Users must also create TIPS Spaces and add their IoT metals (TMOs) to those Spaces.

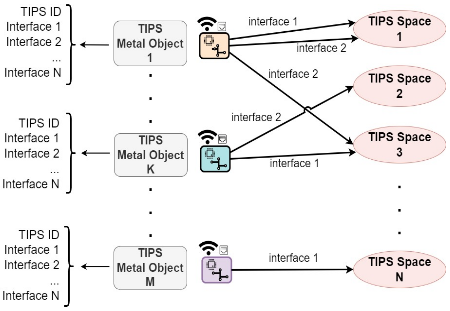


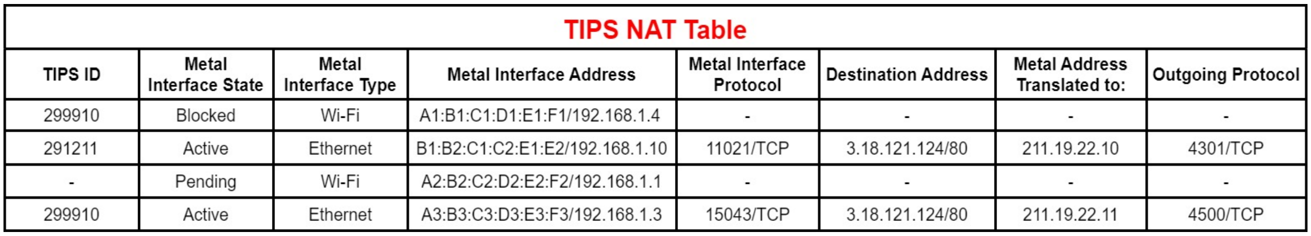
Fig. 3. A high-level view of the IoT metals assignment to different TIPS Spaces.

A TIPS Space is a logical provisioning management entity with a globally unique TIPS ID that enables users to apply software configurations on IoT metals in groups and for different TIPS software provisioning phases. A single IoT metal can belong to one or more TIPS Spaces at any given time (using the same or different network interfaces) among which one TIPS Space must be selected as the default Space for the IoT metal. Default Spaces are always Phase 1 Spaces and TIPS performs software provisioning on IoT metals in Phase 1 only based on the configurations defined in IoT metals default Spaces. Phase 2 and 3 software provisioning can be performed by applying configurations from one or more Phase 2 and Phase 3 TIPS Spaces assigned to the registered IoT metals.

For instance, users can create separate TIPS Spaces for different user programs that they aim to deploy on their IoT metals in Phase 2 and Phase 3 to define their role in an IoT system or environment. If an IoT metal is part of multiple Phase 2 and Phase 3 Spaces, the IoT metal receives user programs and software updates from all TIPS Spaces that it belongs to. This way, TIPS users can better manage and control the provisioning of user programs on their IoT fleet.

It is important to note that TIPS performs provisioning of IoT metals using the same network interface(s) that links them to a TIPS Space. If more than one network interface is used with a single TIPS Space, the TIPS multi-homing is triggered allowing the IoT metals to receive image bits over all registered network interfaces in the Space. We note that Phase 1 provisioning can only use a single network interface that is PXE-compatible. TIPS stores all the information about TMOs (e.g., hardware addresses) and TIPS Space assignments in a data store in TIPS Master that is managed by the TIPS Device and User Management (DURM) service in the Controller.

After TMOs and Spaces are created, the joining process can then begin by IoT metals/devices to start broadcasting Hello messages (Fig. 2, Label 1). Broadcasting Hello messages



TIPS NAT Table

TIPS ID	Metal Interface State	Metal Interface Type	Metal Interface Address	Metal Interface Protocol	Destination Address	Metal Address Translated to:	Outgoing Protocol
299910	Blocked	Wi-Fi	A1:B1:C1:D1:E1:F1/192.168.1.4	-	-	-	-
291211	Active	Ethernet	B1:B2:C1:C2:E1:E2/192.168.1.10	11021/TCP	3.18.121.124/80	211.19.22.10	4301/TCP
-	Pending	Wi-Fi	A2:B2:C2:D2:E2:F2/192.168.1.1	-	-	-	-
299910	Active	Ethernet	A3:B3:C3:D3:E3:F3/192.168.1.3	15043/TCP	3.18.121.124/80	211.19.22.11	4500/TCP

Fig. 4. A view of the NAT table maintained by the TIPS Gateway (TIPS-G).

happens by IoT metals sending DHCPDISCOVER messages that can be captured by a TIPS Gateway (TIPS-G) instance in their environment. For Wi-Fi metals/devices this can happen after they join the Wi-Fi SSID of a nearby TIPS-G instance. Once a TIPS-G instance receives DHCPDISCOVER messages from a new IoT metal, it responds back to the given metal by sending a DHCPOFFER message providing the metal with a private IP address. If the metal accepts the offer, the metal can reply back and asks for configuration using a DHCPREQUEST message. At this point, the given TIPS-G instance replies to the IoT metal by sending a DHCPACK message that assigns the given IP address to the metal's network interface, configures the TIPS-G instance as the network default gateway, and configures the metal to contact the TIPS Master for receiving additional configurations, if the given network interface supports network (PXE) booting.

After configuring the IoT metal's network interface with an IP address, TIPS-G also records the metal's interface type, hardware address, and assigned IP address in its NAT table (see Fig. 4) to use for enabling the metals communication with the TIPS Master and the external networks. At this point, TIPS-G keeps the metal's network interface in the "Pending" state, and shares the recorded information about the metal and its network interface with TIPS-D (Fig. 2, Label 2) for metal registration. Once TIPS-D receives the new IoT metal information, it checks the metal's network interface hardware address with its corresponding TMO in the DURM records to ensure that the metal has an owner (Fig. 2, Label 3) and is pre-authorized for provisioning in Phase 1 (i.e., having a default TIPS Space).

We note that it is possible that a metal is assigned to only one TIPS Space but the Space is not set as the metal's default (Phase 1) Space by the metal's owner. In this situation, TIPS does not allow the metal to proceed to Phase 1. If the metal ownership is confirmed and the metal is pre-authorized for Phase 1 provisioning, TIPS-D confirms the metal's registration for the TIPS Gateway by also providing the metal's globally unique TIPS ID to the gateway (assigned to the metal earlier when its TMO was created by its owner). In response, TIPS-G sets the metal's network interface state to "Active" (see Fig. 4) which allows the metal to communicate directly with the TIPS Master and external networks.

If the IoT metal's information shared by the TIPS Gateway is not confirmed, TIPS-D rejects the registration, informs the gateway of the metal's unapproved join attempt using the given network interface, and also provides the gateway with the TIPS ID of the metal for future references. In response the gateway also blocks the metal network interface from joining again. As shown in Fig. 4, it is possible for an IoT metal to have multiple interfaces that are in different registration states (e.g., Metal ID 299910). TIPS allows policies to be set on individual IoT metals network interfaces separately.

Records of failed registration incidents can be also retrieved from TIPS-D, using the TIPS Master APIs, for processing by AI applications to build anomaly detection models. It is worth mentioning that for security reasons, an IoT metal's network interface cannot transmit traffic to external networks while it is in "Pending" or "Blocked" states. The "Pending" state, however, is temporary and remains in effect for the network interface of an IoT metal while the TIPS Gateway awaits TIPS-D's registration decision.

To remove a network interface from "Blocked" state, a user must first add the interface hardware address to the metal's TMO on TIPS Master and then restart the network interface on the metal. Once a new IoT metal/device interface address is added, TIPS-D checks whether it has the address in its records as it keeps tracking all interactions and information exchange between the TIPS Gateway instances and itself. If the interface address has been previously blocked, TIPS-D releases the lock and informs the corresponding TIPS Gateway instance to clear the record from its NAT table so that the metal can try rejoining using the given network interface.

B. TIPS Multi-Phase IoT Metal Provisioning

In the following, we first introduce the unit of software updates (i.e., Glaze) in the TIPS architecture. We then discuss the three phases of IoT metal provisioning and software configuration in TIPS in more details.

1) TIPS Glaze Images: In TIPS, a workflow image containing all software pieces that configure an IoT metal is divided into separate Glaze images that can be used during separate provisioning phases. A Glaze image, as shown in Fig. 5, is a self-executable binary that is composed of three elements: 1) a name that describes the Glaze image, 2) the binary data that present the image content, and 3) an SHA-256 hash of the image data which is used by IoT metals in different provisioning phases to verify the integrity of the Glaze image data.

Using Glaze images and the TIPS multi-phase provisioning model, we expect that IoT metals can initially receive the

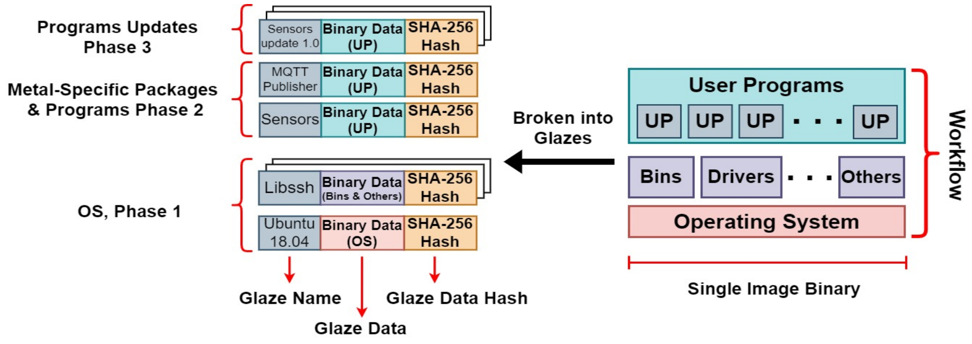


Fig. 5. Workflow decomposition to multiple Glaze images for use in different phases of TIPS multi-phase provisioning model.

minimum software pieces they need to boot and become available to IoT applications (i.e., Phase 1 provisioning) instead of a large workflow image deployment that contains all software pieces needed to configure the IoT metals at once. Once completing Phase 1 provisioning, IoT metals can continue to receive software updates that keeps them updated and compatible with IoT systems and applications needs.

We note that Glaze images must be provided to TIPS Master's Glaze image repository before they can be used by IoT metals. To do so, users can use the TIPS Master APIs to prepare and insert their images. Users can also define which TIPS Space(s) can use a particular Glaze image for provisioning on their IoT metals/devices.

2) Provisioning of Non Configured IoT Metals (Phase 1 Provisioning): TIPS configures raw/bare IoT metals by following the same standard steps that are described in section I. After the initial metal discovery process is completed for an IoT metal and the IoT metal is registered with TIPS-D, the IoT metal can use the IP address of the TIPS Master to connect to the TIPS Master and begin the provisioning process, as shown in Fig. 2, Label 4. Once the TIPS Master receives a provisioning request from an IoT metal, a request handler will be created for the given IoT metal (Fig. 6, label 2) which tasks the Update Manager in the Controller to select the corresponding Glaze image that is intended for configuring the given IoT metal (Fig. 6, label 3). The Update Manager reads the metal information from DURM, selects the corresponding Glaze image based on the metal's default TIPS Space, and informs the Metal Provisioner of the given Glaze Image ID (Fig. 6, label 4). Meanwhile the IoT metal is receiving initial configurations from the Tinkerbell OSIE service [10] (Fig. 6, Label 1). The initial configurations will be used by the IoT metal to prepare an in-memory execution environment for the provisioning of the Glaze images [10].

In this study, we choose to use the open source Tinkerbell hardware provisioner [10] as our IoT metal provisioner engine. We note that the TIPS architecture does not include the hardware provisioner engine that installs the Glaze images on the IoT metals and is also agnostic to the choice of the users for the underlying metal hardware provisioner. Instead, it provides additional services, components, and the mechanisms that guide the provisioning process (e.g., multi-phase provisioning) regardless of the provisioning engine type. We also note that having the ability to use different metal hardware provisioning engines is a design decision that can provide TIPS users with flexibility to customize their TIPS implementation based on their IoT fleet requirements (e.g., scale and heterogeneity). TIPS can also use multiple metal provisioning engines that work in parallel to meet the needs of different IoT metals hardware technology. Some metal provisioners can only work with specific hardware configurations (e.g., ARM CPUs).

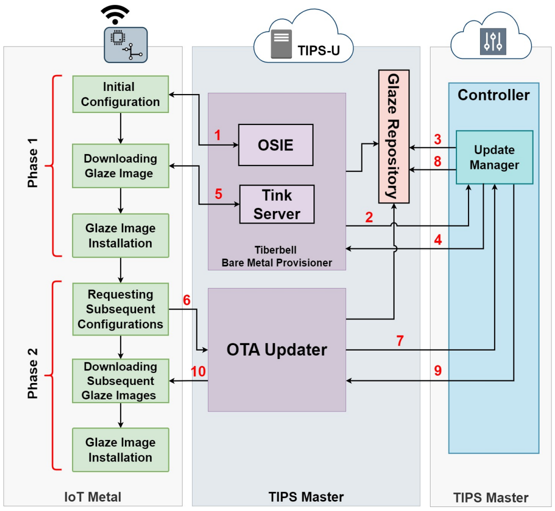
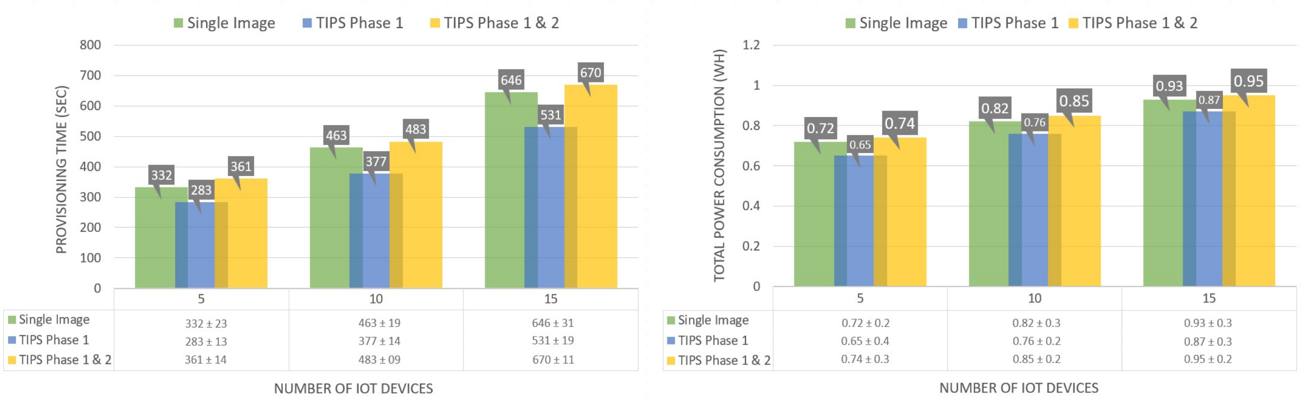


Fig. 6. IoT metal and TIPS-U service interactions during TIPS Phase 1 and Phase 2 of the metal provisioning processes.

Once initial in-memory configuration from OSIE is completed for an IoT metal, the provisioning process continues by the IoT metal to download the Phase 1 Glaze image for installation (Fig. 6, label 5). This step is also handled by the Tink Server as we use Tinkerbell in this study. After the Glaze image is successfully downloaded, the IoT metal installs the image binary and boots from the image. The image installs a full-fledged operating system and all necessary components like drivers, bins, and developer libraries and also includes the TIPS Client (TIPS-C) program which interacts with the TIPS Master to receive subsequent Glaze images intended for the IoT metal in Phase 2 (Fig. 6, Label 6) and Phase 3.



	5	10	15
🟩 Single Image	332 ± 23	463 ± 19	646 ± 31
🟦 TIPS Phase 1	283 ± 13	377 ± 14	531 ± 19
🟨 TIPS Phase 1 & 2	361 ± 14	483 ± 09	670 ± 11

	5	10	15
🟩 Single Image	0.72 ± 0.2	0.82 ± 0.3	0.93 ± 0.3
🟦 TIPS Phase 1	0.65 ± 0.4	0.76 ± 0.2	0.87 ± 0.3
🟨 TIPS Phase 1 & 2	0.74 ± 0.3	0.85 ± 0.2	0.95 ± 0.2

Fig. 7. Average IoT metal provisioning time (left plot) and energy consumption (right plot).

3) Phase 2 Provisioning: by receiving a Phase 2 provisioning request from a TIPS Client, the TIPS Master authenticates the TIPS client with the user private key that is embedded in the Glaze image installed on the IoT metal during Phase 1 provisioning. When a user creates Glaze images for Phase 1, a private key of the user must be included in the image by the image owner which can be later accessed by the TIPS Client on the IoT metals during Phases 2 and 3 provisioning. The public key associated with the private key in the Glaze image is available to the TIPS Master and stored in DURM. We note that users can choose different key pairs for different images they install in Phase 2 and 3 and can change their keys after Phase 1 provisioning is completed.

After successfully authenticated using the associated keys, TIPS allows the IoT metal to proceed with Phase 2 provisioning given the IoT metal has been authorized to obtain a Phase 2 Glaze image. This happens by the metal's owner user to pre-assign the IoT metal to one or more Phase 2 TIPS Spaces. If the IoT metal is assigned to more than one TIPS Space, it receives multiple image installation during Phase 2 provisioning (i.e., each with its associate keys). Needless to say that failures in authentication and authorization of the IoT metal for Phase 2 provisioning can prevent the metal to proceed to Phase 2 and Phase 3 provisioning.

If the IoT metal proceeds to Phase 2 provisioning, a new request handler will be created by the TIPS Master for the IoT metal's request (Fig. 6, label 7) which tasks the Update Manager to select the subsequent Glaze images that must be delivered to the IoT metal for installation (Fig. 6, label 8). The Update Manager reads the IoT metal information from the DURM and selects the corresponding Glaze images based on the metal's TIPS Space configurations. The Update Manager also shares the selected Glaze images IDs with the TIPS-U OTA Updater service (Fig. 6, label 9). Next, the IoT metal receives the Glaze images from the OTA Updater (Fig. 6, label 10) and installs them using the TIPS Client on the metal. The TIPS Client also acknowledges the success of image installation on the metal to TIPS-U.

4) Performance Evaluation: We evaluated the performance of TIPS multi-phase IoT metal provisioning model in terms of average provisioning time and energy consumption level in Phase 1, Phase 1 and Phase 2 combined, and compared it against using a single full-fledged image that contains all the software pieces installed during Phase 1 and Phase 2.

The experiment includes a 4GB workflow image for use in the single image provisioning experiment and three Glaze images of the same workflow image with 1) a 3GB image of the operating system and all its software components for Phase 1 provisioning and 2) two 500MB Glaze images that contain user-specific packages and programs for Phase 2 software provisioning. As stated before, we used Tinkerbell as the TIPS hardware provisioning engine and Raspberry Pi 4 Model B with 8GB RAM as our IoT metals. We also used a server with CORE i5 CPU and 8GB of RAM as the TIPS Gateway (TIPS-G). The TIPS Master is placed in the cloud using a virtual machine with 16 vCPUs and 16GB of RAM. All Raspberry Pi 4 devices in the experiment connect to TIPS-G using a high-performance Gigabit Ethernet switch.

We repeated our experiments ten times (i.e., reporting average values) and used different number of IoT devices each time to measure scaling impacts. As the results in Fig. 7 suggest, the average total provisioning time (left plot) for provisioning of IoT metals in Phase 1 and 2 is only marginally higher (i.e., 3-8%) than the provisioning of all software pieces in a single image installation. Meanwhile, IoT metals can become on average 15-18% faster available to IoT applications by passing only Phase 1 software provisioning compared to a single image workflow installation. This amount of time, which on average is also more than one minute in many cases, can let IoT metals to receive another round of authentication and authorization before being programmed to perform actual IoT systems tasks, which can consequently improve the availability and timeliness to respond to security changes in different IoT systems.

We also observed in our results that using TIPS, the average IoT metal provisioning duration increases as the number of IoT metals scales. This is mainly because of using TIPS-G as the default gateway for all traffic to outside networks and the TIPS Master, which can sometimes limit the available bandwidth when multiple IoT metals undergo parallel software provisioning. To address this issue, we suggest that users may use multiple TIPS-G instances in an IoT system or use higher capacity links that meet their infrastructure size. It is worth mentioning that the size of the Glaze images used for provisioning and the choice of Phase 1 provisioning engine can also play important roles in observing higher or lower device provisioning times at larger systems.

In terms of average power consumption (Fig. 7, right plot), the results show the same pattern of marginal increase as a result of TIPS going through Phase 1 and Phase 2 of provisioning compared with a single image installation of all software pieces used in Phase 1 and Phase 2 combined. We note that power consumption is typically a function of time, workflow load, and the metal hardware specifications and can consequently vary for different workloads and various IoT settings. The intention here, however, is to use Raspberry Pi 4 devices to demonstrate that the TIPS multi-phase metal provisioning approach can be used in IoT settings with minimal energy consumption overheads. As also observed, the energy consumption during Phase 1 provisioning is expectantly lower compared with combined Phase 1 and Phase 2, or the single workflow image provisioning.

5) Phase 3 Provisioning: After Phase 1 and 2 provisioning are completed, TIPS enters Phase 3 provisioning in which fully configured IoT metals from Phase 2 can continue to receive software updates (e.g., security patches and program updates) until they remain registered and active. Unlike Phase 1 and Phase 2 in which IoT metals initiate the software provisioning process, in Phase 3, the TIPS Master notifies registered IoT metals when there is a new software update available for them. The process involves the TIPS Client on the IoT metals to subscribe to TIPS Master for Phase 3 updates after Phase 2 provisioning is completed. During the subscribing process, TIPS verifies the identity and authorization of the IoT metals, once more, by using the associated key pairs for the metal's programs used for Phase 2 to ensure that only authorized IoT metals can receive further software updates.

Successful IoT metals get notified when there is a Phase 3 update available in any TIPS Phase 3 Spaces that they are registered. Phase 3 update notifications also include instructions on how to obtain the software updates (i.e., Glaze image IDs, update schedule, etc.) from the TIPS Master. The TIPS Master may also require additional authentication and authorization to deliver new updates to IoT metals if no authentication occurred for a given IoT metal in a past settable period of time (e.g., 120 seconds).

C. Security Benefits of TIPS Multi-Phase Provisioning Model

Unlike current IoT metal provisioning and life cycle management systems that install the entire software stack of an IoT metal at once, TIPS, as stated before, splits IoT metal provisioning in separate phases where each phase begins with an authentication and authorization step with failures in authentication and/or authorization to prevent IoT metals to be provisioned in that and subsequent phases. We believe that the TIPS approach can provide several key security benefits and advantages that improve the overall security of IoT metal provisioning and life cycle management.

For instance, the TIPS approach can increase the timeliness (responsiveness) of reactions to changes in IoT systems security requirements. IoT systems like smart transportation, smart grids, and utility networks have tight security requirements which can change depending on the system conditions. Provisioning of new IoT metals/devices is a timely process which can introduce delays in applying updated and new security measures and policies. Using the TIPS multi-phase provisioning model, IoT metals can become available much sooner to IoT systems and applications and new security policies can be applied faster on new IoT metals, even before they can obtain the programs that define their roles in the system.

The TIPS multi-phase provisioning and authentication and authorization model can also prevent IoT systems from unauthorized OS installation and mass deployment of malicious software stacks on IoT metals/devices in IoT systems. Verifying the metals access to workflow images and the programs they receive during each provisioning phase can prevent malicious software stack deployment on the IoT metals/devices in vulnerable and compromised IoT systems. One feature of the TIPS Client software installed during Phase 1 provisioning on IoT metals is its ability to ensure that no program can be installed on IoT metals/devices without being provisioned via the TIPS Master. If a program is installed manually (after Phase 1 provisioning), the TIPS Client informs the TIPS Master which subsequently blocks all the metals interfaces to external networks.

III. Related Work

A. IoT Metal Hardware Provisioning

There are important IoT metal hardware provisioners that offer software provisioning services equivalent to the combined Phase 1 and 2 of the TIPS software provisioning model. Razor [11], Tinkerbell [10], TOSKA [12], and Foreman [13] are all examples of such important IoT metal provisioners. Among these provisioners, Foreman can provide a comprehensive life cycle management of IoT metal hardware including workloads installation and monitoring of the IoT metal resources after provisioning [13], [14]. TOSKA, is primarily designed for service provisioning, but the same as Foreman, it is also capable of provisioning and automating IoT metals in various IoT systems and environments [12].

Tinkerbell allows iPXE-compatible IoT metals [15] to boot and fetch pre-defined workload images from the TinkServer [10]. To communicate with the TinkServer, an IoT metal must obtain the TinkServer address from Tinkerbell Boots service during network booting [10]. Tinkerbell is a powerful metal hardware provisioner and can support wideranging IoT metals. Razor [11] is also very similar to Tinkerbell but IoT metals that use Razor for provisioning must also use a customized Razor Microkernel image that initializes the metals and enables them to download and install workflow images [11].

Phoenix [16] is an IoT metal configuration management system similar to TIPS that leverages metal hardware provisioning services of an underlying provisioning engine (e.g., Tinkerbell, Razor, etc.) to offer IoT metals life cycle management. Phoenix main focus is on the reconfigurability of edge IoT metals and can only provide a one-time single workflow image provisioning with no authentication of the IoT metals performed during the provisioning process [5].

IoT-compatible bare metal provisioning services are also provided by cloud provisioning engines like the OpenStack Ironic [17], [18] and MAAS [19]. Ironic and MAAS are both very powerful provisioning engines that are mainly intended for use in cloud datacenters and for bulk hardware configurations in large IoT systems and environments. In compare, TIPS and other aforementioned metal provisioning and life cycle management systems are more ideal in smaller scales IoT systems.

B. Over-the-Air Update Systems

Similar to IoT metal hardware provisioners, there are several over-th-air update systems that can manage the life cycle of IoT metals configuration. Mender [20] is a powerful example of such systems that provide OTA software updates to wide-ranging IoT metals. Mender's software updates can include both the metal firmware and its applications [20] and it can be easily integrated with U-Boot, an open-source boot loader that is commonly used in embedded devices to perform low-level hardware initialization [21].

Mender can also be used to apply more complex IoT metals configurations and software updates such as changing an IoT metal operating system or its kernel. To account for failures, Mender usually entertain updates on an auxiliary storage (e.g., Flash memory) or a separate file system partition before they are final. Similar to Mender, SWUpdate [22], [23] and RAUC [24], [25] are also well-known systems that provide OTA software updates to embedded devices such as IoT metals among which RAUC is designed to be very lightweight with the main binary file to be less than 1MB in size. Over-the-Air software updates are also commonly provided by cloud-based platforms for IoT and embedded systems [2], [26] with transportation and connected vehicles being the trending use cases [27].

IV. Conclusion and Future Work

We proposed TIPS as a new cloud-based IoT metal provisioning and life-cycle management system. TIPS can provide a multi-phase IoT metal provisioning model that (a) makes IoT metals available to IoT applications faster during the provisioning process, (b) improves the timeliness of provisioning systems to respond to security changes in IoT systems, and (c) provides a secure IoT metal provisioning experience with minimal overhead in terms of provisioning duration and power consumption. TIPS can support IoT metals provisioning over wired and wireless network interfaces and has the ability to support multi-homing. In the future, we are interested to measure TIPS performance when multi-homing is used.

Acknowledgments

This work was supported by the Canada Excellence Research Chair in Intelligent Digital Infrastructures at the University of Toronto, funded by the The Tri-agency Institutional Programs Secretariat.

References

[1] J. Bauwens, P. Ruckebusch, S. Giannoulis, I. Moerman, and E. De Poorter, "Over-the-air software updates in the internet of things: An overview of key principles," *IEEE Communications Magazine*, vol. 58, no. 2, pp. 35–41, 2020.

[2] M. M. Villegas, C. Orellana, and H. Astudillo, "A study of over-the-air (ota) update systems for cps and iot operating systems," in *Proceedings of the 13th European Conference on Software Architecture-Volume 2*, 2019, pp. 269–272.

[3] S. Quincozes, T. Emilio, and J. Kazienko, "Mqtt protocol: fundamentals, tools and future directions," *IEEE Latin America Transactions*, vol. 17, no. 09, pp. 1439–1448, 2019.

[4] A. Chandrasekar and G. Gibson, "A comparative study of baremetal provisioning frameworks; parallel data laboratory," 2014.

[5] M. Johnston and S. Venaas, "Dynamic host configuration protocol (dhcp) options for the intel preboot execution environment (pxe)," Tech. Rep., 2006.

[6] W. Croft and J. Gilmore, "Rfc0951: Bootstrap protocol," 1985.

[7] R. Droms, "Rfc1541: Dynamic host configuration protocol," 1993.

[8] D. K. R. Sollins, "The TFTP Protocol (Revision 2)," RFC 1350, Jul. 1992. [Online]. Available: https://www.rfc-editor.org/info/rfc1350

[9] G. Malkin and A. Harkin, "Rfc 2349: Tftp timeout interval and transfer size options," Tech. Rep., 1998.

[10] "Flexible automation for bare metal — tinkerbell.org," https://tinkerbell.org/, [Accessed 11-06-2024].

[11] "GitHub - puppetlabs-toy-chest/razor-server: Razor is next generation provisioning software that handles bare metal hardware and virtual server provisioning — github.com," https://github.com/puppetlabs-toy-chest/razor-server, [Accessed 12-06-2024].

[12] T. E. M. M. F. Steiler, "Bare-metal provisioning of internet of things devices by means of tosca."

[13] "Foreman — theforeman.org," https://theforeman.org/, [Accessed 12-06-2024].

[14] K. Vonblohn, "Centralizing server and workstation provisioning, configuration, and management with foreman and puppet." in *Proceedings of the 2021 ACM SIGUCCS Annual Conference*, 2021, pp. 23–25.

[15] "iPXE - open source boot firmware [start] — ipxe.org," https://ipxe.org/, [Accessed 12-06-2024].

[16] M. Moghaddassian, S. Shafaghi, P. Habibi, and A. Leon-Garcia, "Phoenix: Transformative reconfigurability for edge iot devices in small-scale iot systems," *IEEE Access*, 2023.

[17] "OpenStack Ironic Bare Metal — openstack.org," https://www.openstack.org/use-cases/bare-metal/, [Accessed 12-06-2024].

[18] C. G. Kominos, N. Seyvet, and K. Vandikas, "Bare-metal, virtual machines and containers in openstack," in *2017 20th conference on innovations in clouds, internet and networks (icin)*. IEEE, 2017, pp. 36–43.

[19] "Metal as a Service — MAAS — maas.io," https://maas.io/, [Accessed 12-06-2024].

[20] "Over-the-air software updates for IoT devices — Mender — mender.io," https://mender.io/, [Accessed 12-06-2024].

[21] "The U-Boot Documentation &x2014; Das U-Boot unknown version documentation — docs.u-boot.org," https://docs.u-boot.org/en/latest/, [Accessed 12-06-2024].

[22] M. Mäkipää, "Comparison of ota update frameworks for linux based iot devices," Master's thesis, 2022.

[23] "Home — swupdate.org," https://swupdate.org/, [Accessed 26-06-2024].

[24] "RAUC — rauc.io," https://rauc.io/, [Accessed 12-06-2024].

[25] L.-C. Duca, A. Duca, and C. Popescu, "Ota secure update system for iot fleets," *International Journal of Advanced Networking and Applications*, vol. 13, no. 3, pp. 4988–4992, 2021.

[26] P. P. Ray, "A survey of iot cloud platforms," *Future Computing and Informatics Journal*, vol. 1, no. 1-2, pp. 35–46, 2016.

[27] S. Halder, A. Ghosal, and M. Conti, "Secure ota software updates in connected vehicles: A survey," *arXiv preprint arXiv:1904.00685*, 2019.