Enabling Secure IoT Systems Using Mbarx

A Platform Approach to Securing the Internet of Things

Contents

Figures

Introduction: What is IoT?

The term IoT is broadly used to describe a system of distributed devices deployed on connected networks, controlled by application logic and monitored by analytics. Conceptually, IoT is similar to networked devices and sensors that have been around for decades, however, legacy systems had the luxury of addressing security and connectivity by using private networks with known infrastructure.

Today, IoT device deployments are anticipated to exceed 20 billion by 2020, a figure that is largely driven by the ubiquity of public Internet infrastructure. This change to the paradigm means that it is no longer possible to own the end-to-end network. Instead, today's connected devices need to withstand deployment in unknown environments, route mission critical data securely, across hostile networks and do it all in a fast and reliable way.

Arcturus has developed Mbarx as a simplified, secure approach to IoT systems. This document will present an overview of Mbarx Secure IoT, the platform architecture, elements, security and performance.

IoT Protocol Primer

Mbarx Secure IoT was developed to address the connectivity and secure communication challenges that exist with IP based IoT systems. Existing protocol standards in this space, including MQTT and CoAP are inherently insecure. These protocols are primarily intended for message passing and do not consider the underpinning connectivity or management functions of the end-point itself. Additional effort and resource, on top of the core message passing system, is required to consider the application of a security architecture, firmware handling, maintenance and configuration. When using one of these standards, it is important to weight the necessity of this type of message passing service. Not all applications benefit from complex message queuing and delivery techniques, thus the overhead of validating this protocol layer, then enhancing it, can unduly delay work on the core IoT application itself.

Other industry protocols such as LWM2M, a superset of CoAP, make security a core function. This increases the overall size and complexity of the system and restricts users to implementing the OMA standard set of management objects. This is ideal for IoT applications that want to maintain cross-compatibility, but may not suit the need of every specialized IoT application. MQTT, CoAP and LWM2M meet specific implementation needs, as do HomeKit, AllJoyn, Thread, Weave, Brillo and Fuscia. Selecting the right solution will depend on what the application is, who is going to use it and to some extent, what is supported by the cloud provider. As HomeKit is not intended for industrial control monitoring, Mbarx is also not intended to be used on a smartphone tethered fitness tracker. The power of Mbarx is in IP network connectivity, using industry standard TLS security. The elegance of Mbarx is a simple socket protocol for management and operational control of end-points.

Mbarx System Elements

Mbarx consists of end-point nodes, tools and gateways. These elements work together as a platform that can be built upon as required:

- Mbarx End-point nodes
  - IoT devices consisting of network connected embedded micros, running an RTOS or Linux.
- Mbarx Tools
  - PC based applications (Windows® or Mac®) for developers and administrators.
  - Tools support site-wide management functions including configuration and firmware deployment.
- Mbarx Gateways
  - Edge devices running one of several Mbarx gateway solutions.
  - Gateway solutions are available to support access, connectivity, IoT site services or interactive workflow.

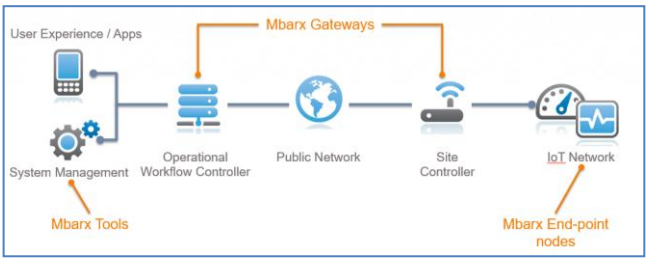

Figure 1 - Mbarx End-points, Tools and Gateways

Mbarx End-points

The Mbarx end-point stack consists of two components; a multicast discovery announcement and a point-to-point protocol. The discovery announcement advertises basic information about the end-point using the Multicast Domain Name Service (MDNS) protocol. The MDNS announcement does not pass beyond a local network, but does provide enough information for other devices, tools or systems to identify each unique device and establish a point-to-point TCP connection – in effect, a pairing

service. The TCP connections are handled securely, requiring TLS. Once connected, a higher-level command protocol communicates across the socket connection to lower-level system functions via internal middleware interface integrated into the OS. This combination of secure point-to-point protocol and device discovery represents the core of the Mbarx end-point connectivity and security architecture. Mbarx end-point middleware provides:

- Hooks into the RTOS or OS to configure and manage the network stack e.g.:
  - DHCP, static networking, network fall-back schema
  - NTP, remote system logging
- Defined objects for device credentials e.g.:
  - ID, name, operating mode, location, device type, firmware type
- Configuration and operation of core application e.g.:
  - Application specific objects and workflow
  - I/O and other peripherals
  - Pass-through communication to UART and other peripherals
- System monitoring e.g.:
  - Connectivity heuristics, link state, network fall back operation
  - Supervision of application connectivity, monitoring and alarms
  - Watchdog and logging

Mbarx end-point software is available for Linux, MQX or FreeRTOS and can be enabled on devices with as little at 256Kbyte of internal flash and 128Kbyte of SRAM.

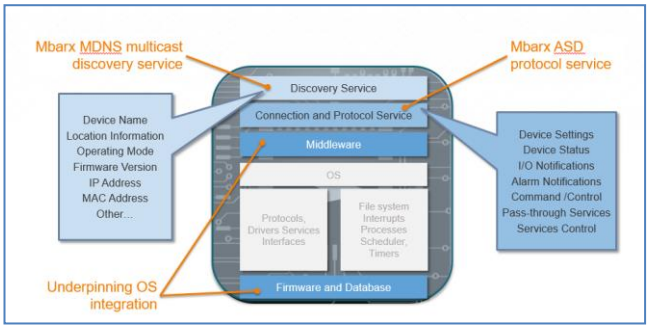

Figure 2 - Mbarx End-point Software

Service Modes

Different networks have different requirements. A commercial HVAC system implemented in an office building may exist on a private network and not require external access. In contrast, a home IoT thermostat exists on an unknown network, behind a firewall and needs reliable cloud connectivity to provide value to the user. This latter type of deployment poses a connection problem as the cloud server cannot create a connection to the device. The device must first egress the network and "call home" to the cloud server. Mbarx end-points do this by using a remote server connection facility to automatically originate a connection on reboot, periodically or on demand. This connection can remain persistent, keeping the cloud service active/live, or it can terminate after inactivity to reduce traffic and maximize security. The connection is supervised with notifications and alarm conditions reporting any unexpected loss of the host connection. Since this facility is a generic connectivity service for the Mbarx protocol, once the connection is established any monitoring or device management can then be conducted.

End-point Security

Transport Layer Security (TLS) is the industry standard authentication and encryption method used for securing TCP/IP communication. Mbarx uses TLS v1.2 with an AES256 (optional Diffie-Hellman) cryptographic cipher and SHA256 certificate with 2048-bit RSA key authentication. However, TLS is only the start of an overall security architecture. In addition to TLS, the Mbarx end-point architecture supports:

- **Authenticity** - by verifying firmware payload is code signed
- **Integrity** - by ensuring transmitted firmware payload data is complete and unaltered
- **Compatibility** - by ensuring device and firmware types match

These additional protections are there to help minimize threat vectors and improve system reliability. Device specific implementations can further enhance this performance by providing dual-redundant flash image storage, image failover, flash read lockout and tamper detection. Combined with Mbarx tools and gateways these security features form an end-to-end, chain-of-trust architecture.

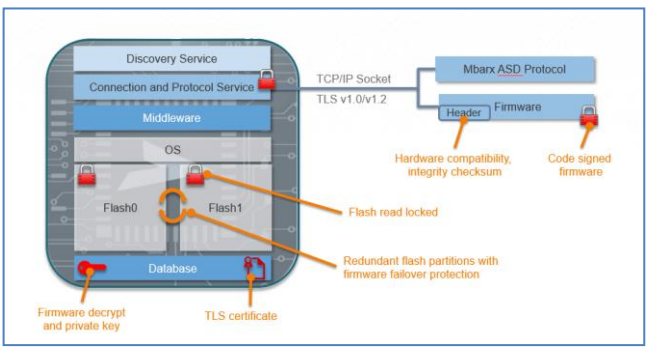

Figure 3 - Mbarx Security Architecture

Failsafe Operation

Mbarx end-points operate in either an *autonomous* or *controlled* mode. In *autonomous* mode events trigger actions to form basic workflow logic, without the need for an external host application. In *controlled* mode, the device relies on an external host application for command and control. Events are transmitted via the Mbarx connection either by using the secure TCP/IP socket or a dedicated UART connection. The host application supervises the device and provides the workflow logic.

An often-overlooked aspect of reliability is detecting failure and failing in a deterministic way. The Mbarx end-point stack addresses this by:

- In *autonomous* mode
  - A supervising application can passively monitor the autonomous device behaviour.
  - The supervising application can intervene under special conditions, provide parallel processing logic or log tasks with higher fidelity.
- In *controlled* mode
  - The socket is supervised.
  - If the connection is lost, the device will report an alarm condition and can optionally fallback to an defined autonomous behaviour.

Firmware Updates

Firmware updates follow the security architecture outlined in the end-point security section, supporting TLS secure communication, plus payload authenticity, integrity and compatibility. Firmware is determined to be compatible prior to download and payload is inspected prior to any execution. Linux systems handle this by probing in RAM, prior to writing to flash. MCU systems, due to limited SRAM, write the firmware payload to an inactive, redundant flash partition, then probe in place. Initializing parameters are only changed after qualifying the firmware is from a trusted source and has not been manipulated. If the device fails to boot, it is recovered using a watchdog service. After which the next update is then written to the partition where the failed firmware is located and subjected to the same scrutiny prior to use. This provides a high-assurance boot, based on an always-working firmware image.

Mbarx End-point Protocol (ASD)

Mbarx ASD is the high-level, point-to-point protocol used for both configuration and operational control of an end-point. ASD runs over either a secure TCP/IP socket or a dedicated UART interface which provides a low-complexity, request/response character interface into the underlying system. ASD commands are available to get or set virtually all system parameters.

A development tool called the Mbarx Virtual Control Panel is available to help become familiar with ASD, debug or experiment with workflow. The tool is provided as a PC application and connects to a device across a network connection. A source code package (written in Python and QT) is available, combined with support, to help developers who may want to use it as a starting point for a custom application.

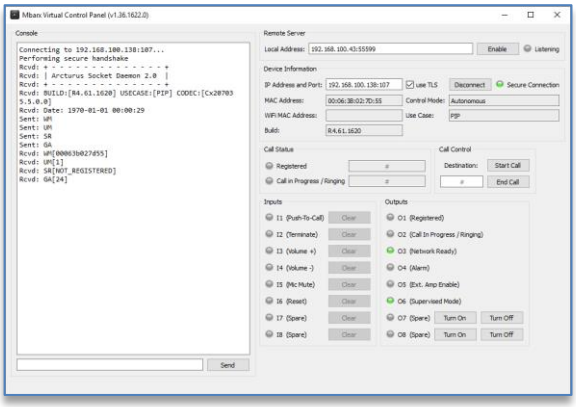

Figure 4 - Mbarx Virtual Control Panel

Creating Custom Application Logic using Mbarx

The Mbarx ASD protocol is used to create custom application logic. This can be done using the network connection to support cloud-host applications or by a UART connection to support a local-host application. The network approach allows cloud application logic to control many end-points across a system, while the UART approach creates a way to integrate with an existing system or write application logic on a connected MCU. Adding a low-cost MCU has proven to be a cost-effective way to enable custom application logic or extend overall capabilities. MCU selection is left up to the preference of the developer, helping to reduce the implementation learning curve. Extensions in the ASD protocol make it possible to pass-through data between the network and UART connections, making it possible to use this facility to remotely update the MCU application code.

Integration with External Peripherals

In Mbarx, access to board-level peripherals is accomplished using ASD protocol extensions that pass-through data between network and peripheral interfaces e.g.: UART, SPI or I2C. These pass-through extensions enable the network addressing of connected peripherals such as sensors, actuators or specialized equipment to provide a higher-level of

system integration. Since the pass-through communication occurs as part of the Mbarx ASD protocol connection, it is therefore subject to the same rigorous security across the network. Consolidating this communication into a single Mbarx data stream prevents the need to establish additional socket connections, open ports and supervise sessions. Specialized devices or use cases can be accommodated easily via a support engagement.

Mbarx Tools

Mbarx tools include two PC based applications available for Windows or Mac – an IoT site management tool called System Manager, and a Virtual Control Panel tool, used mostly during development to become more familiar the ASD protocol and for workflow experimentation, as described in the Mbarx End-point Protocol section.

System Manager

Mbarx System Manager is used for all aspects of device configuration and maintenance, either on a local LAN or by connecting to a remote site. System Manager is an administrator-level tool that presents a list of devices and supports click-through workflow for configuration and maintenance tasks. It can support concurrent management actions on many devices at the same time and is verified to support over 100 simultaneous firmware update tasks. Granular status progress is fed back to user and visual confirmation is provided once settings are changed. System Manager supports the Mbarx security architecture, using TLS, along with the following key features:

- Configuration and configuration templates
- Firmware updates
- Reset (reboot), factory reset (restore)
- Firmware store, to securely acquire firmware
- Firmware repository, to locally save firmware
- Auto-upgrade sever for “call home” feature
- Connectivity to remote sites

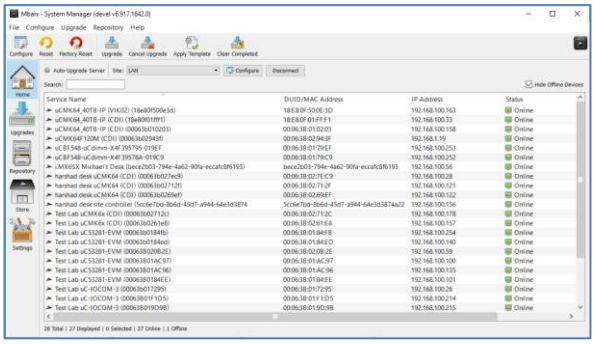

Figure 5 - System Manager Home Panel Showing Device List

Firmware Store and Repository

System Manager supports a method to securely acquire firmware images from a trusted source and an internal firmware repository, stored on the local machine. The System Manager firmware store provides a method to connect to Arcturus cloud servers providing access to the most recent firmware image for any Arcturus platform. Firmware is downloaded into the System Manager repository, which is stored locally on the PC hard drive. The local repository is accessible regardless of connectivity, making it convenient for installers who may be at sites where internet connectivity is not available. Custom firmware can be manually added to the repository through an import function.

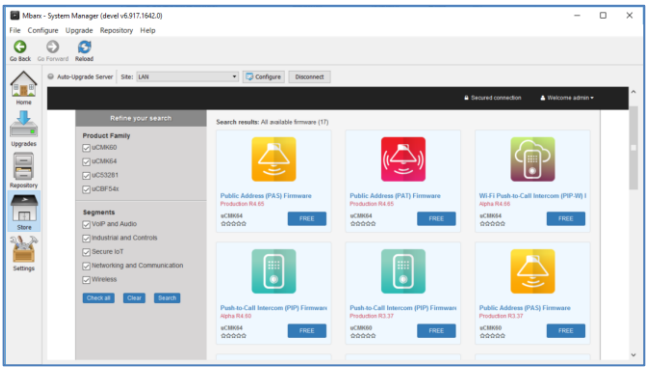

Figure 6 - System Manager Store with Available Firmware Downloads

Auto-Upgrade Server

The auto-upgrade server provides a method for devices to connect to System Manager and obtain firmware updates automatically. This process is initiated by the end-point’s remote server connection facility, which allows the end-point to “call home” to a predefined address. Internal to this facility are features that allow the end-point to establish this connection to on a periodic basis, after reboot, or on demand. Once the end-point has securely connected to System Manager upgrade server, the end-point will provide the server with its hardware attributes and firmware version. System Manager compares these parameters to the firmware records stored in its local repository. If a compatible firmware version is older than the version that is flagged in System Manager as the current version, then System Manager initiates an update.

Remote Sites

Out-of-the-box, System Manager is a powerful tool for securely managing end-points on a local network. System Manager *remote sites* extends this paradigm by providing a method to securely connect to an Mbarx IoT Gateway, located at the edge, or inside a

remote network. Setting up a remote site requires that the FQDN or IP address of the Mbarx Gateway is added as a remote site in System Manager. Once added, switching between sites is accomplished by using a drop-down box to select between LAN and each remote site. By selecting the site, System Manager establishes a connection to the remote Mbarx Gateway. Once connected, System Manager obtains the profile of the remote site from the gateway. This site profile is presented to the user in the same way a local network site is presented, providing access to all the same features from the same common System Manager interface regardless of site location. Mbarx Gateways can also provide additional connectivity and site services; these services can also be configured using System Manager.

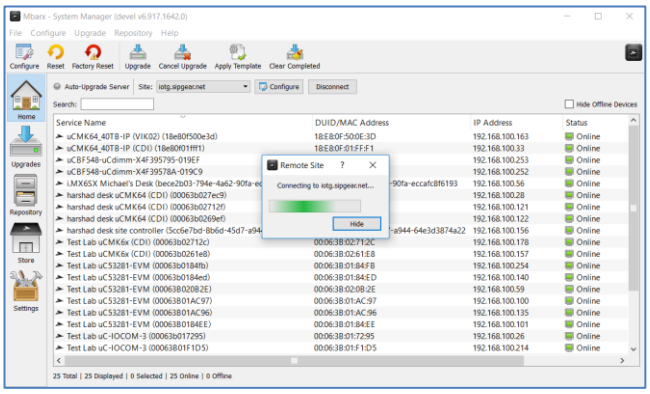

Figure 7 - System Manager Connecting to Remote Site

Mbarx Gateways

Mbarx Gateways are physical devices that provide various services to an IoT network. Appropriate gateway hardware will depend on the requirements of the application; Arcturus has three options for IoT Gateway hardware, these range in performance and features including:

- NXP P1020 – Dual-core, Power architecture, communications-class processor
- NXP i.MX6 – Cortex-A9 application processor with media support (pre-production)
- NXP LS1012A – Cortex-A53, 64-bit networking processor (pre-production)

The following Mbarx IoT gateway solutions are available for the above platforms:

- Mbarx Operations Controller, for interactive workflow systems
- Mbarx Site Controller, for remote site connectivity and services

Mbarx Operations Controller

The Mbarx Operations Controller gateway stack is intended for developing secure interactive workflow systems. It is suitable for various types of applications where user supervised workflow is required, including security and access control, nurse call, patient care or mass transit management systems. The core of this solution is built around a hierarchical grouping and notification system. Devices are associated to a location; locations are then grouped hierarchically. This forms a continuum, e.g. rooms, floors, buildings etc. The notification system is tied to events that are fed from devices, these events are routed to subscribed users through the Operations Controller framework or externally via push notification, SMS, VoIP, Twitter, or other method.

A mobile responsive html5 front-end provides the workflow interface, this allows users to jump seamlessly from notification to operational control regardless of where they are in the world or type of device they are using. The user interface provides the ability to tie in other html objects such as live video, historical information, location, account data etc. The system is multiuser, highly configurable and supports user and supervisory workflow interfaces.

Mbarx Site Controller

The Mbarx Site Controller gateway stack supports secure connectivity and IoT site services. The Site Controller maintains information about devices on the IoT network by listening for discovery announcements. This database forms the profile of the IoT site and is provided to System Manager when it connects to the gateway. This method allows System Manager to support the remote site in the same way a local site is supported, providing the same capabilities, using the same click through workflow. In addition, custom IoT cloud applications benefit from this connectivity by having improved reliability and an end-to-end secure data path to the IoT site, via a single aggregated connection.

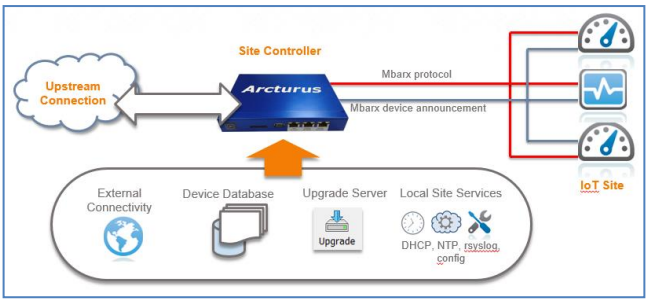

Figure 8 - Site Controller Basic Services

In many environments the gateway would separate connectivity between the IoT site and the internet, isolating each network on discrete interfaces and supporting edge-network and gateway services such as firewall, secure access, whitelists, NTP, DHCP, etc. This model is sufficient for IoT networks where the gateway's external facing interface is publicly addressable (such as in a DMZ). Often however, multiple subnets or NATs may exist between the IoT site and a publicly addressable interface. System Manager itself is likely also in a similar environment, behind one or more NAT or subnet. In this case, more advanced connectivity can be provided by using a Site Controller set up in a *Domain Controller* mode. In this mode, the Domain Controller acts as a master for other Site Controllers. These Site Controllers register their location with the Domain Controller and it tests and maintains the connections to the remote sites. The Domain Controller can relay payload data between host applications and remote sites, if required. This method traverses complex network topologies without using vulnerable protocols such as dynamicDNS or UPNP or blindly leaving ports open. The Domain Controller uses an independent method to authenticate with other Site Controllers, allowing relayed connections to be established directly between the two TLS session participants, without the intervention of Domain Controller. This protects payload data security by eliminating any need for the Domain Controller to have access to session credentials or any unencrypted payload data.

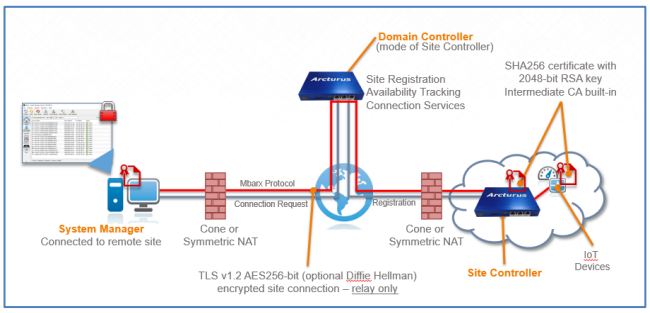

Figure 9 - Site Controller Advanced Connectivity

Site Controller Services

Much like System Manager, Site Controllers can provide a firmware update service for the local IoT network. By using System Manager for configuration of the Site Controller, it is possible to push a firmware image to the remote Site Controller's repository, then, using the same method described in the Auto Upgrade Server section, IoT end-points can connect to the Site Controller and obtain the most recent firmware version. Hosting firmware on the gateway eliminates the need for many devices to egress their home network, thus reducing bandwidth, vulnerability and improving update reliability. The Site Controller performance is qualified to handle up to 300 concurrent updates using Arcturus IoT Gateway hardware. In addition to connectivity and firmware services Site Controllers can provide network services, such as DHCP, NTP and rsyslog or specialized services, such as voice/PBX, network bridging/bonding, failover or 3G/LTE redundancy.

Implementing an Mbarx System

Mbarx is intended to provide the core building blocks of IoT system infrastructure. This platform approach provides security, connectivity and management functions without defining the core IoT application (data payload). Mbarx components including, tools and gateways can scale or get added as an IoT deployment grows and evolves. Arcturus offers a complete range of services to help support and customize hardware and software, including OEM branding options.

Demos and Availability

Development kits are available for various types of end-point devices and gateways. An online lab is available for demonstrations of Mbarx including end-points, System Manager and Mbarx Site Controller capability. Contact Arcturus for a free demo or view one of our online tutorial videos. Visit Arcturus at https://www.arcturusnetworks.com/ or learn more about Mbarx by visiting the product page https://www.arcturusnetworks.com/products/mbarx/ or connect with Arcturus directly by email or telephone +1.416.621.0125 x233.

CONTACT INFORMATION

Arcturus Networks, Inc., the Authors of this document can be contacted at:

Arcturus Networks Inc

701 Evans Ave. Suite 300

Toronto, ON

Canada

M9C 1A3

URL: www.ArcturusNetworks.com

TEL: +1 416.621.0125

TEL: +1 866 733 8647

FAX +1 416.621.0190

COPYRIGHT NOTICE

NOTICE OF MARKS

ARCTURUS NETWORKS INC. - LIMITATION OF LIABILITY, INTENDED USE.