SD321605

Equipment Monitoring and Control Through Forge Viewer

Bandu Wewalaarachchi
Eutech Cybernetic

Learning Objectives

- Discover security risks and why two-way communication with equipment via ALG (application level gateway) is important
- Learn about the limitations of IoT platforms, and gain knowledge of available technologies that you can use to implement two-way communication
- Learn about what it takes to communicate with conventional building systems

Description

IoT (Internet of Things) is a term that is used under many variations. It could be an addressable IP device on a network or a device that supports non-IP based communication protocol (such as ZigBee), which is indirectly connected to the network as an IP node through some hub. In both cases, device is assumed to be intelligent enough to report its status on its own. Also, it is assumed that the device is intelligent enough to securely identify genuine commands coming from authorized users or systems.

This is not at all the case with legacy devices and control systems that are deployed in the buildings today. This class intends to make audience understand the difference and what it takes to implement real-time communication with legacy devices & systems. Then, extend it to allow monitoring and control of devices via Forge Viewer.

This class will be using Eutech Cybernetic's Lucy iPasS service as infrastructure for simplicity although it is not mandatory for implementing real-time communication with Forge Viewer. It will also include short demonstrations of 'iviva Smart BIM' solution that is built on top of Autodesk Forge as a use case.

Speaker

A technical expert with extensive experience in software design and development, integrated solutions across SCADA, ERP, BIM and Cloud Services, and management of IT personnel, in the fields of facility and infrastructure management, industry automation, BIM and enterprise software integration especially in the areas of smart workplace, smart cities, Internet of Things and mobility. Inventor of patented technologies.

Thought leadership and specialties: Understanding technology trends and applying them to products & services, Managing technical people, Intellectual property management, Communicating technology matters to non-technical people, Technical team building, training, coaching and mentoring of technical staff

Background

IoT – Future

There is no argument about potential of IoT to change the world and the people in it.

With the increasing need of flexibility to rearrange building systems to experiment with ideas and to allow incorporating user feedback, it is desired that devices stay loosely attached to the systems. IoT is perfect for this.

Another reason for the popularity of IoT is its promise to be 'do-it-yourself' item, although not all of them are there yet. Once they become easy to handle, supplying and fixing an IoT device to extend a building system would be far more cost effective than retrofitting a conventional building subsystem.

Similar to how smart phone's camera takes photos, scans documents and reads QR codes, a single IoT device has potential to contribute to multiple systems simultaneously. That would be another strong reason why they can be highly effective in building systems.

Technically speaking, IoT also has potential for distributed processing without making a central system a bottleneck for growth and expansion. For example, a CCTV network with intelligent cameras do not burden central system to pull and analyze images from all cameras. Instead, individual camera will process its own image stream and report important information (metadata) to the central system for decision making.

Another advantage of IoT is their ability to self-diagnose and report, which would enable true predictive maintenance. It could lower lifetime cost of systems.

Two-way communication with IoT

Similar to definition of the term 'IoT' being wide, IoT-platform is also a term with a wide definition. While a few solutions are able to provide two-way communication with the devices (for monitoring and control), many IoT platforms only collect the data transmitted by IoT devices and make them available to users or to analytics.

Security with IoT

There are two concerns of security when it comes to IoT.

First, the device being on a public network increases its vulnerability to external-hacking. Security of IoT highly depends on IT-security measures that are put in place. More specifically, managing of SSL certificates in individual devices is an important task when managing an IoT network. Some of IoT platforms provides functionality for IT personnel to handle this centrally.

Secondly, the risk of hijacking an IoT device is perhaps a bigger security concern. Since it is a 'computer' on its own, it could become a Trojan horse.

These concerns make IoT less suitable for small enterprises until IoT platforms grow to tackle all security concerns.

Cost of IoT

Cost of IoT is not just the manufacturing cost of the device but the entire workforce required to look after them. Difference becomes contrasting when a system requires a large number of devices – say, multi story building that needs a few thousands of addressable lights.

Legacy devices – Today

Masking devices with Application Level Gateway

Most of building systems are still made up of legacy devices that are not suitable for facing IP networks. Some of them lacks IP compatible protocol support. Some of them do communicate through IP networks but do not have adequate security to guard against potential misuse.

As such, communicating with both IoT and legacy devices/systems require a similar architecture.

Application Level Gateway (ALG) acts as a ‘reverse proxy’ when communicating with devices. It forms a bridge between the private network carrying devices and the Internet (or the office network).

This architecture allows gateway to receive commands through Internet through a secured channel and ‘validate’ them before sending them to the device to execute.

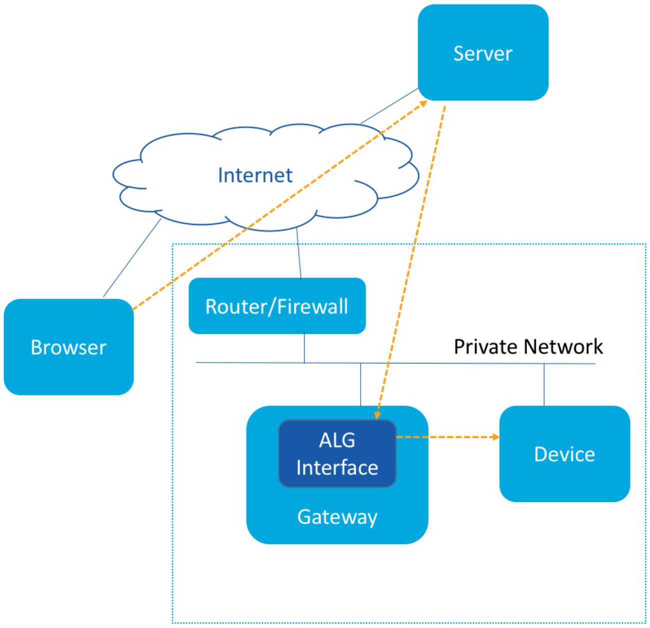

Figure 1 - Safe deployment architecture for Devices

Legacy control system architecture

Most of legacy control systems are made up of three layers:

- sensors & actuators,
- field controllers and
- group controllers.

Role of the sensor is to detect environment parameters such as light intensity, temperature, vibration or stress. Often, sensor provides an analog output by means of a varying voltage or current: more specifically, 0-5V or 4-20mA.

Similarly, actuator performs a physical activity based on an input signal given by means of a varying voltage or current.

Field Controller is responsible in converting this analog value into a digital value (and vice versa). Digital value could be a binary or number representation. One field controller is often connected to many analog devices and they are addressed using the 'port' where analog device is connected.

Most of the field controllers do not communicate via IP networks. Some of them do, but they offer a simple protocol to read values and send commands, which is manufacturer specific.

Role of Group Controller is to connect with multiple Field Controllers and provides a high-level communication channel via IP network. They offer industry standard communication protocols such as bacnet, OPC and Modbus.

Limitations of Controllers

There are limitations of both Field and Group Controllers when handling host communication. First, there are limitations to the payload. Host is not able to ask the controller to 'give me all data'. Instead, host needs to segment the request and send it part by part.

Secondly, controllers need a 'breathing room' between consecutive commands. As such, controllers cannot be polled for data continuously. They need to be ready to execute commands when user needs them to.

Some controllers are able to accept a 'subscription' for 'CoV events' by the host. In this case, controller promises to notify the host in case of a 'change of value'. However, this mechanism has two practical problems hence they are not used in serious systems.

First, most of the group controllers have a limit of active subscriptions. That means, host is able to subscribe to only a set of points – not all of them. Secondly, when a controller malfunctions and do not send any notifications, host would be simply assuming that no 'change' to the values of points that it subscribed.

On-demand real-time communication

Most common practice with legacy devices and subsystems is communicating with them on-demand – ask for values when there is a need to. Need for the new value could be that user may want to check status, or system wants to periodically check them for alarm evaluation or historical data capture.

Making Requests by Browser App

On-demand communication becomes challenging with when the Browser becomes the client software.

When browser makes a request to a host (server) via http, results are not returned in-line with the request but return data is received later, which is known as asynchronous network request.

On top of that, when server has to send the data request to the gateway to fetch from a device, server will not receive a response in-line with the request but it will arrive later, asynchronously.

Dataflow of this transaction is shown in the figure 4 below.

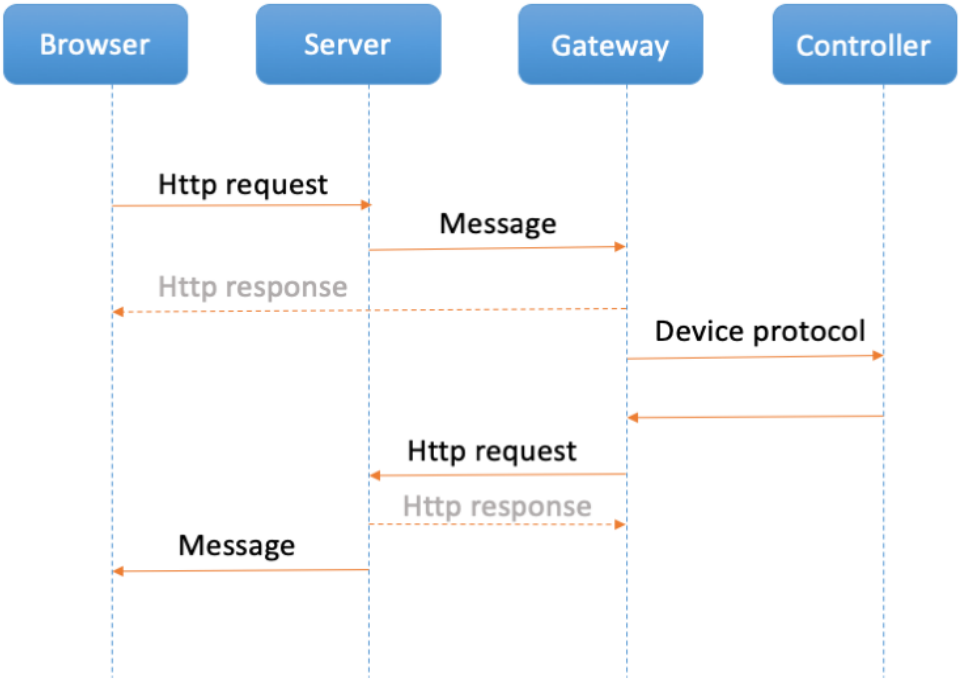

Figure 4 – Communication dataflow

In this architecture, Brower makes a http request to server by specifying 'device address'. Device address must include necessary information to reach the device, namely, gateway address and device address within the gateway.

Coding level concerns

Since the message carrying the payload arrives asynchronously and independently to the http request, it would be necessary to implement a mechanism to correlate the response with the request.

This could be accomplished by sending a 'cargo' (some token) together with the request and have the gateway return that together with the response.

Simpler approach

This problem can be simplified by using an integration platform such as 'iviva Lucy service' where it can hold the http response at the server until the gateway send the payload through a message. Then, include that payload in the http response (as shown in the figure below).

This dramatically simplifies the code at the browser level that integrates real-time communication with Forge Viewer.

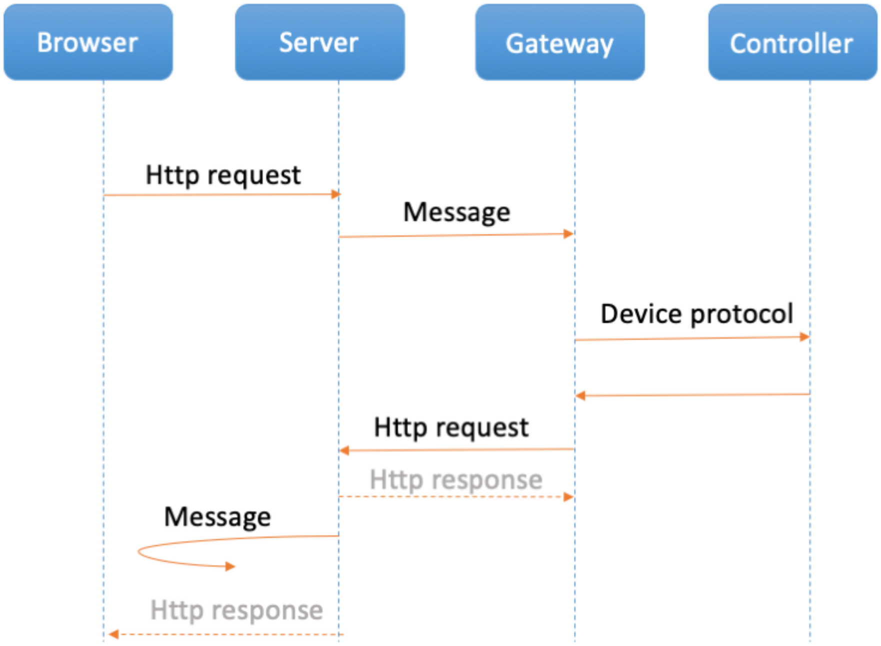

Figure 5 - Simplified real-time communication with a proper integration platform