

Gateways for the Industrial Internet of Things: Emerging trends

Industrial Internet gateways help translate among competing, closed silos of proprietary stacks that provide vertical integration between the “things” in the field and the integrated cloud services that make them useful.

Key concepts 

- The IIoT trend toward fragmentation is at odds with the needs of system integrators and manufacturers.
- Software portability significantly improves the development speed and cost of connected systems.

Remote connections and new software licensing structures can create opportunities for incremental upgrades in control and automation software, advancing efficiency, improving information flow, and bringing greater optimization. The confluence of ubiquitous, open standard-based Internet connectivity and powerful, low-cost embedded devices has led to the emergence of the Internet of Things (IoT).

A projected 25-billion IoT-connected devices are expected to be online by 2020. This trend disrupts the established control and

instrumentation field, opening it up to a new range of possibilities referred to as the Industrial Internet of Things (IIoT).

The development of the Internet itself has been characterized by collaboration, interoperability, and conformance to open standards in contrast to the foundational ethos of the Internet. In a manner foreshadowed by proprietary mobile phone ecosystems, the IoT/IIoT space is emerging as a new “wild west” in which competing, closed silos have emerged. Each silo is vying to lock in a maximal share of the expected huge market by providing a

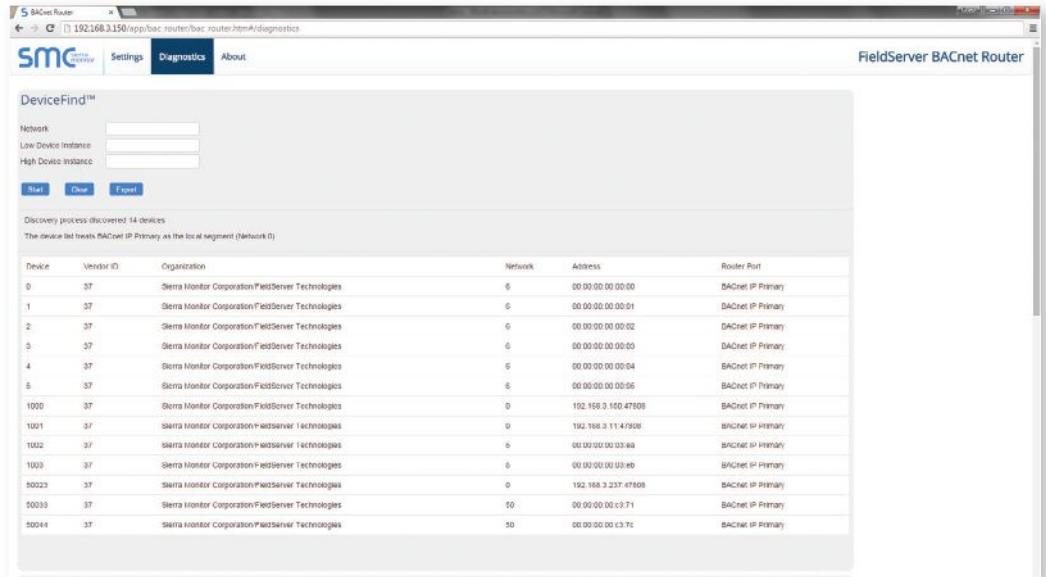


Figure 1: The FieldServer BACnet Router from Sierra Monitor Corp. is used by system integrators to interconnect BACnet networks using different media, such as RS485 field networks and IP networks, typically to connect the devices to a building management system. While BACnet routing is a standardized, low-level function, the FieldServer IIoT gateway architecture, on which the BACnet Router is based, affords it the ability to host application-level functions that add value for system integrators, removing the need for other tools to verify that all expected devices can be reached and identified. Images and graphics courtesy: Sierra Monitor Corp.

full, closed, proprietary stack providing full vertical integration between the “things” in the field and the integrated cloud services that make them useful.

The IIoT trend toward fragmentation is at odds with the needs of system integrators and manufacturers in control, instrumentation, building management, and energy management fields. Over the past two decades, these fields have been characterized by industrywide collaboration in the creation of open standards that have exposed suppliers to more direct competition, while at the same time expanding their market reach. The result has created significant benefits to the industry as a whole. The convenience, added value, and efficiency of this approach is under threat from the new silos.

In reaction to this danger, a number of vendors have formed the Industrial Internet Consortium (IIC, www.iiconsortium.org) to work toward establishing a consensus around interoperability.

To this end, the IIC has published the “Industrial Internet Reference Architecture Technical Report” (IIRA), which sets out an architectural framework intended to guide the development of Industrial Internet Systems (IIS).

The IIRA explores architectural concerns from business, usage, functional, and implementation viewpoints. The implementation viewpoint discusses the technology and communication schemes required by the reference architecture and is of special interest for the topic at hand.

The IIRA outlines a number architecture patterns that largely depend on the presence of gateway devices, such as:

- Three-tier architecture pattern with edge, platform, and enterprise tiers

This pattern uses a gateway to connect the edge tier (containing field devices or edge nodes linked by a proximity network) to the platform tier (containing data transforms, oper-

ations, and analytics) via an access network, which provides for data and control flows to be exchanged. Figure 7-2 from the IIRA illustrates this pattern.

- Gateway-mediated edge connectivity and management architecture pattern

Here an edge gateway provides the link between a local area network (LAN) of edge devices and a wide area network (WAN) linking to higher level services. The edge gateway may

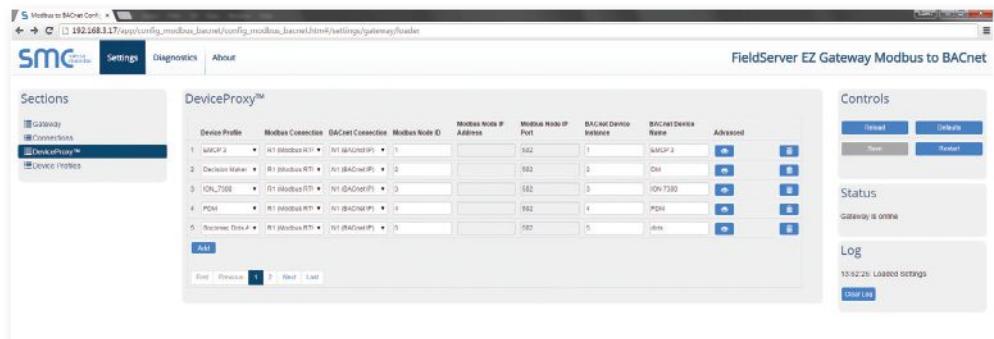


Figure 2: The FieldServer EZ-Gateway from Sierra Monitor Corp. illustrates a standardized approach to field device virtualization. In this example, a system integrator creates a profile that configures data items to be read from a Modbus device and mapped into a BACnet data model, which allows the user to enrich the data model with: object name, object type, units, and description fields. Each profile can then be used to create multiple DeviceProxy instances corresponding to multiple Modbus devices in the field. The image below shows four DeviceProxy instances being instantiated and mapped into virtual BACnet devices.

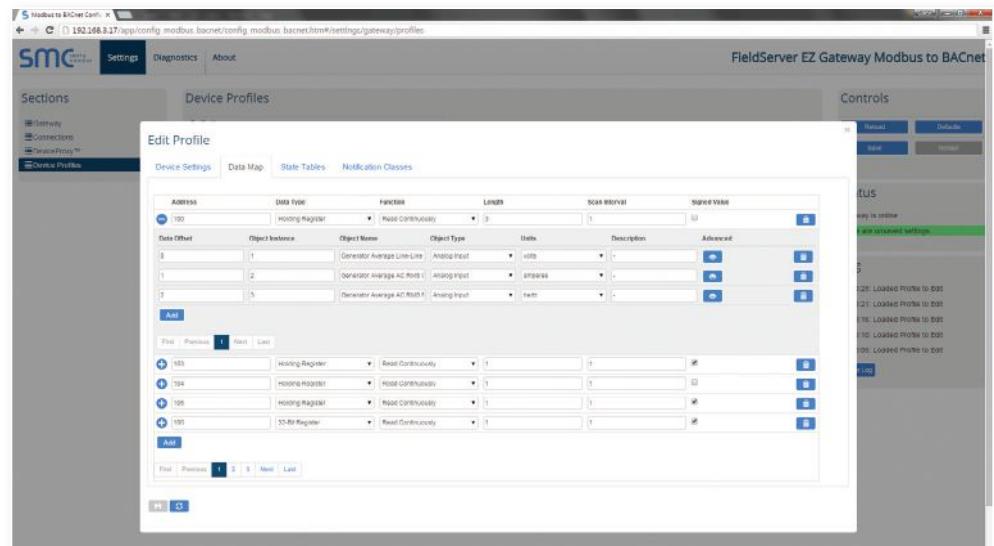


Figure 3: A portable data format by which device profiles that have been manually configured on a gateway can be exported and shared with users via a profile sharing website hosted by the gateway vendor. A publicly available library of labor-saving profiles are indexed by a field device model can be uploaded and instantiated on compatible gateways by any system integrator.

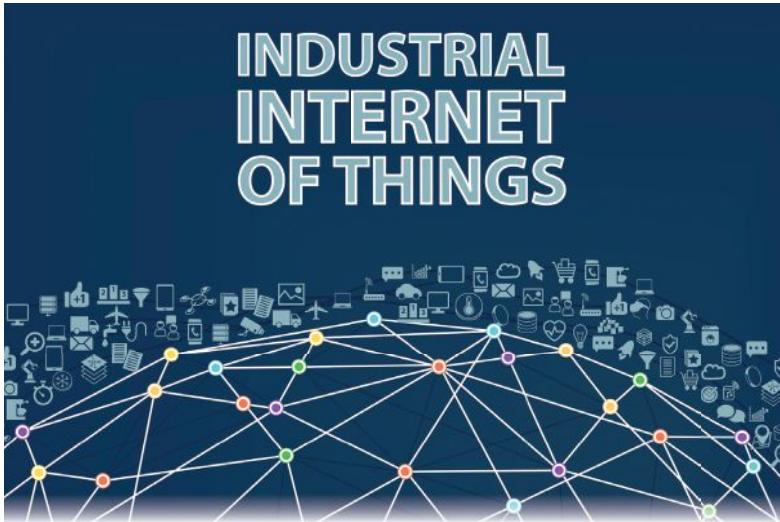


Figure 4: Growth in devices connected to the Internet disrupts the established control and instrumentation field, opening it up to a new range of possibilities referred to as the Industrial Internet of Things (IIoT).

itself act as the provider of local connectivity by acting as a hub. It serves to insulate the edge devices from the WAN and can contain some data processing, analytics, control logic, and application entities.

As the widespread acceptance of modern, open-field protocol standards has reduced the need for traditional gateways in the field, the IIoT has created a need for a new breed of intelligent gateways that unlock the full potential of interoperability among diverse real-world devices and industrial Internet systems.

Data management

Industrial and building automation gateways have moved far beyond the simple message translation paradigm of the classical definition of gateways. Gateways can now occupy all levels of the open systems interconnection (OSI) networking model. By supporting application layer entities, modern gateways can actively read, write, and manipulate data, as well as implement intelligent data caching, data logging, and controller functions. Store-and-forward techniques allow for the maintenance of continuous historical records where Internet connectivity is unreliable or intermittent.

Virtualization, abstraction

Beyond caching and mirroring data, gateways are becoming important providers of data abstraction. By modeling diverse field devices in consistent, self-documenting, virtual-device models, gateways insulate the higher tiers of

the system against the countless different data structures encountered in the field. Similarly, field device functions can be wrapped and presented in a coherent manner. For example, a gateway can render a collection of very diverse energy meters in an abstract representation that is consistent across an IIS, thus, insulating higher tiers of the system from often unavoidable variations in the field.

Network management

Gateways are often ideally located to host field network management functions, removing the need for third-party network management tools.

Network discovery, self-configuration

Taking the network management concern a step further, IIoT gateways are becoming capable of zero-configuration deployment. Once such a gateway has been installed in a field network, it is able to detect field devices autonomously and consolidate data and functions into an integral interface.

Crowdsourcing

The enormous fragmentation and diversity of the existing, installed base of products can make some degree of human interaction unavoidable. Many legacy field protocols have no means of describing the data they present, forcing vendors to supply separate documentation of the data maps for each device model. These in turn need to be manually translated into gateway configurations for the data to become presentable in a more integral virtual-device model.

In a new development, a gateway vendor has created a portable data format by which device profiles that have been manually configured on a gateway can be exported and shared with the broader user community via a profile sharing website hosted by the gateway vendor. This is giving rise to a growing, publicly available library of labor-saving profiles that are indexed by a field device model and can be uploaded and instantiated on compatible gateways by any system integrator.

Application hosting

Multi-tier architectures often require a range of functions to be available to local operators even while the site is disconnected from the Internet, or perhaps in instances where a given deployment remains isolated. Additionally, a robust system design might require certain monitoring and control functions to be located locally to maintain uninterrupted operations where Internet connectivity is intermittent.

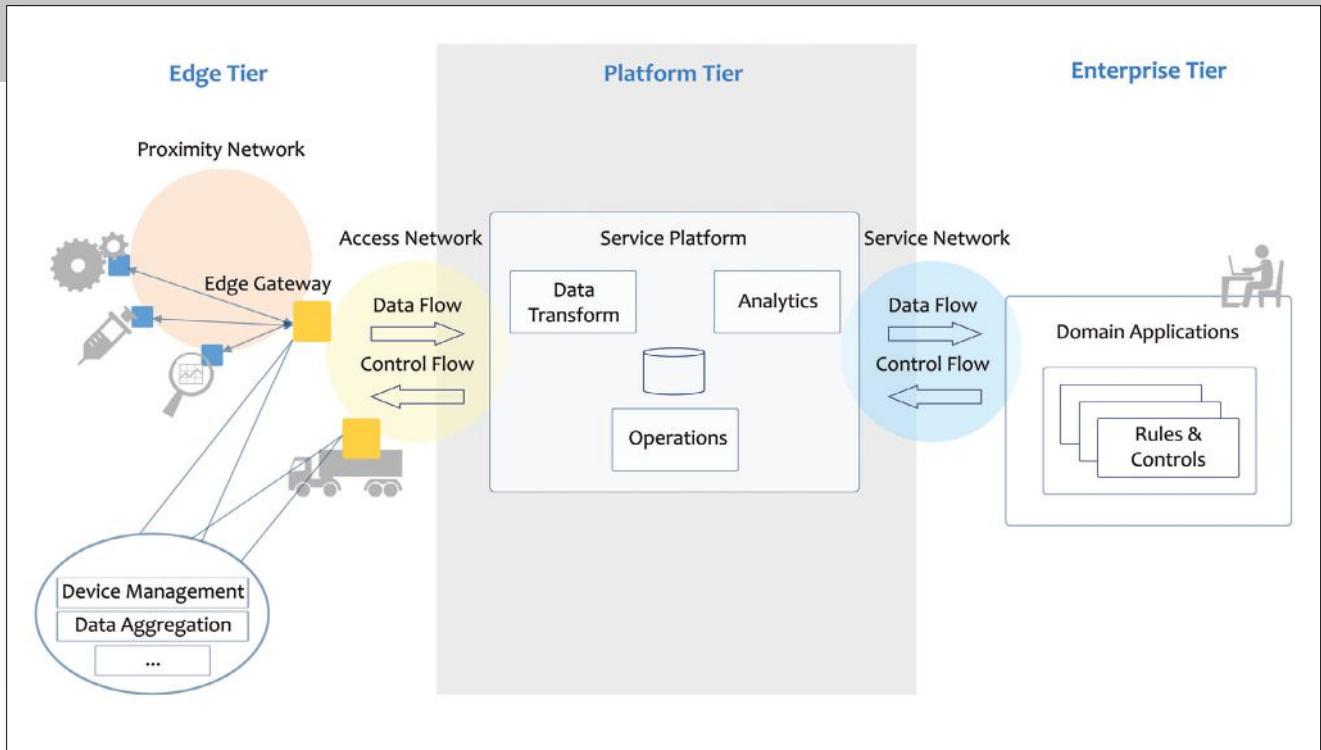


Figure 5: Industrial Internet Consortium (IIC) published the “Industrial Internet Reference Architecture Technical Report,” intended to guide the development of Industrial Internet Systems (IIS). This IIRA Figure 7-2 from the report shows a three-tier architecture pattern with edge, platform, and enterprise tiers. The pattern uses a gateway to connect the edge tier (containing field devices or edge nodes linked by a proximity network) to the platform tier (containing data transforms, operations, and analytics) via an access network, which provides for data and control flows to be exchanged.

Remote access

Gateways are able to host remote access servers, providing secure tunneling access to remote users of applications hosted on the gateway. This minimizes the functionality required of the cloud infrastructure and extends the usability of locally hosted applications.

Gateway architecture developments

The expanded scope of gateway functionality places enormous demands on the productivity and flexibility of the supporting technologies. Where gateways might traditionally have succeeded as monolithic embedded applications programmed in C or C++, the functional diversity and flexibility required today necessitate platforms that allow the desired combination of services to be assembled from a range of custom, purpose-built components and third-party, proprietary, or open-source components. The growing convergence between traditional server environments and embedded environments is creating important opportunities to expand the capabilities of gateways.

Portable software platforms

Software portability significantly improves the development speed and cost of connected systems by enabling communications or application code to be reused on multiple platforms. Two factors have emerged that have enabled

code sharing between edge and cloud platforms: compatible virtual machines (VMs) and portable operating systems.

Compatible virtual machines

For the past two decades Java has made code portable by providing compatible VM implementations for platforms of various sizes. While making comparatively heavy performance demands, today’s edge devices are well able to run full-featured Java VMs and can draw on an enormous code base. Scripting languages, such as Perl, Python, PHP, and more recently, JavaScript, lend themselves to portable implementations because only the interpreter needs to be ported to any new platform. JavaScript is already the world’s most popular programming language, and the rapidly developing Node.js ecosystem offers a very promising portable platform that allows code to be shared between embedded devices, servers, and web browsers. The asynchronous, event-driven nature of JavaScript code makes it well suited to developing I/O-intensive, transaction-driven systems. In addition, npm, an open source package management system (npmjs.com), is delivering unprecedented growth in the number and scope of small, focused, and high-quality modules supporting a vast range of popular protocols, services, application programming interface (API) clients, and database clients and servers.

Table: Modbus data map

Data address	Data value (available via Modbus)	Description (not available via Modbus)
30000	65	Inside temperature in Fahrenheit
30001	59	Outside temperature in Fahrenheit
<more values>		
31000 <last value>	80	Humidity in % relative humidity
31001		<a new value can be added here, at the end>

Portable operating systems

Intruding on the historical preserve of proprietary embedded multitasking operating systems, Linux has emerged as a popular option for embedded devices. Through the RT-Preempt patch, the Linux kernel supports soft, real-time systems and is typically sufficient for field protocols with millisecond-scale timing requirements. Historically, edge devices have used custom, lightweight Linux distributions that were often different from desktop or server counterparts.

Canonical's Ubuntu is a coherent Linux offering covering IIoT operating system needs from edge to cloud through the addition of Snappy Ubuntu Core. Snappy provides a robust, secure, and modular update and a rollback mechanism for embedded devices and cloud-deployed servers alike.

Interoperable, composable software modules

Apart from the use of portable operating systems and virtual machines, architectural flexibility is enhanced by software design practices that encapsulate functional modules in interoperable and composable ways. This generates more options for system designers to decide whether to locate application, monitoring, or control code on an edge gateway or in the cloud, and it maximizes efficiencies through code reuse. In addition to offering internal and downstream abstractions, differences between various upstream cloud-tier providers can be abstracted and wrapped in standard component interfaces, enabling a given gateway implementation to be integrated into a variety of systems with minimal effort.

Rich, self-describing data models

Beginning with XML, and increasingly through the more compact and browser-friendly JSON, gateways can present data and APIs in formats that make sense by themselves, that are

inherently extensible, and that can be integrated and interpreted by many systems.

For example, a traditional Modbus data map would contain large number, 16-bit data registers that are identified by numerical addresses. Such a map would typically be hard-coded into a device, and the only possibility of extending the map would be through appending new values at the end of the map. (See Modbus data map table)

All this information can be expressed in one JSON object, which can be extended without breaking compatibility with existing implementations by simply adding fields, as shown by the highlighted added field, Setpoint, which could simply be inserted without affecting the integrity of the communications. The object is transmitted as a text string and is intelligible on its own. While there is a price to be paid in that the JSON format is much more verbose than Modbus, the advantages greatly outweigh the costs in most cases.

JSON Data Object

```
{
  "Inside Temperature":{
    "Units":"Fahrenheit",
    "Value":65,
    "Setpoint":63
  },
  "Outside Temperature":{
    "Units":"Fahrenheit",
    "Value":59
  },
  "Humidity":{
    "Units":"%RH",
    "Value":80
  }
}
```

Consolidation, maturation

The IIoT ecosystem is at an early stage, and a lot of consolidation and maturation lie ahead. The emerging breed of IIoT gateways will contribute toward containing uncertainty and preserving the value of existing investments by providing rapidly adaptable capabilities for keeping up with a changing environment. In their new guise, as envisaged by the industrial Internet architectures set out in the IIRA, IIoT gateways will remain an indispensable part of the industrial Internet for the foreseeable future. **ce**

- Varun Nagaraj is the CEO of Sierra Monitor Corp. Jens Eggers is a software engineer at Sierra Monitor Corp. Edited by Eric R. Eissler, editor-in-chief, Oil & Gas Engineering, eeissler@cfemedia.com.

Go Online

More information and images on this topic are in the online version. If reading the digital edition, click through the headline; if reading print version, search the headline at www.controleng.com.

Consider this...

As IIoT/IoT develops, many companies are going to try to grab as many dollars as they can by making proprietary software, resembling prior technology wars, such as Betamax vs. VHS and Blu-ray vs. HD DVD.