ADVANTECH
Enabling an Intelligent Planet

Whitepaper

WISE-PaaS Introduction

The success of IoT is stepping away from vertically-oriented, closed systems towards open systems, based on open APIs and standardized protocols at various system levels.

Louis Lu
Embedded-IoT Architect

Agenda

Introduction

This white paper introduces Advantech WISE-PaaS architecture for the Internet of Things (IoT), including devices field-side and server-side, and the Cloud architecture (WISE-PaaS) required to interact and manage the devices. The objective is to provide IoT software developers the basic concepts needed to more effectively achieve IoT solutions. WISE stands for "Wu Intelligence Solution Embedded", Wu is Chinese for the sound that reflects Fog and Things.

This white paper includes:

- An overview of the Internet of Things
- Requirements for IoT Architecture
- The Advantech WISE-IoT Architecture
- The WISE-PaaS Open Software Architecture
- Advantech Edge Intelligence software diagrams
- Interoperability for IoT
- Conclusions

The Internet of Things includes multiple different vertical markets, including

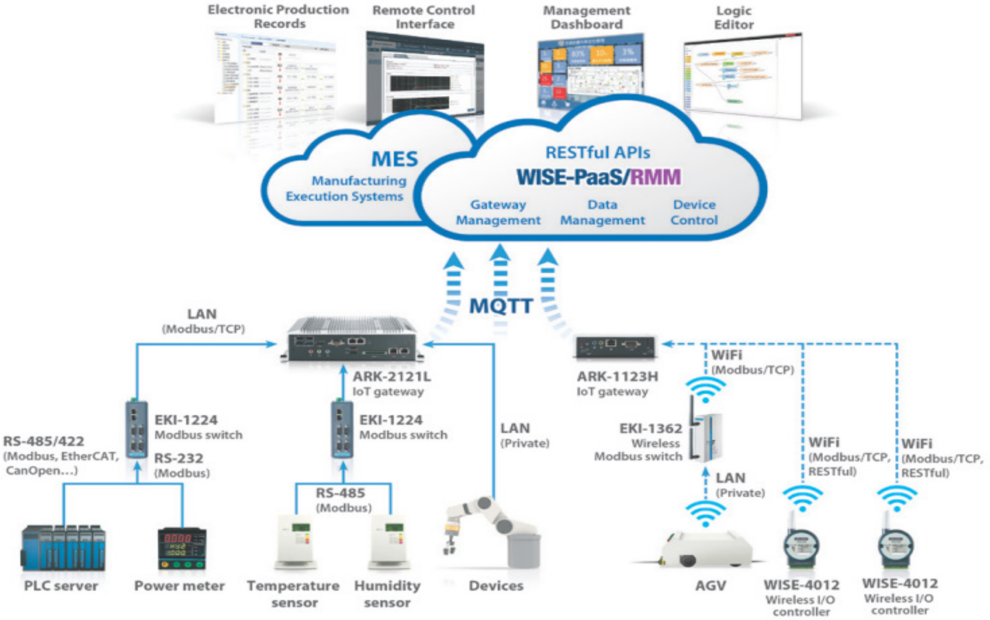
- Factory Automation
- Intelligent Retail
- Transportation
- Agriculture
- and more…

The WISE-IoT architecture does not directly fulfill the above vertical market requirements. It is however a flexible and scalable architecture that supports plug and play capabilities, and supports many functions across a wide variety of use cases is inherently valuable and useful. Advantech proposes the WISE-IoT architecture, which includes many aspects, including the SaaS, PaaS for Cloud service, and also allows developers to manage, monitor, access and process data from IoT devices, the connectivity between the server-gateway or gateway-sensor, and the WISE-agents and software on the devices.

The WISE-IoT architecture that Advantech proposes can support ARM/x86 systems without being specific to a set of technologies; it is highly integrated with the most popular open source projects and technologies. We also provide the APIs and source codes to help the developer to easily implement more and more IoT devices in the field as well as edge/ gateway environments.

An Overview of Industrial Internet of Things

The Internet of Things refers to the set of wireless/wired devices and systems that interconnect real sensors and actuators to the Internet. For examples, the smart meter for green energy, stress lighting control for transportation, environmental detection for smart cities, etc.



Network and Information Protection Options

The number and diversity of devices that are aggregating data will generate explosive growth. By IDC's count, that number is already approaching 200 billion. And the number of sensors (e.g., the accelerometer in your smartphone) that track, monitor, or feed data is already more than 50 billion, with scientists talking about trillion-sensor networks within 10 years. Of course, not all of those 200 billion things are actually wired and communicating on the Internet, but some 20 billion are. And, by 2020, this number will grow by 50% to 30 billion connected devices.

There are six key functions in the IoT solution:

1. Interaction with local IoT devices: devices contact via a short or long range wireless interface such as Bluetooth, ZigBee, RFID, and 6LoWPAN that is responsible for acquisition of observations and their forwarding to remote servers or edge computers for analysis and storage.

2. Local data real-time analysis and processing of sensor data acquired by IoT devices. Most of the gateway platform has powerful computing ability that can execute some intelligent actions on the local side.

3. Interaction with remote IoT devices, directly over the Internet or more likely via a gateway. This is responsible for acquisition of sensor data and forwarding meaningful data to remote servers (or Cloud) for analysis and big data storage.

4. Application-specific data analysis and processing. This runs on an application server (or Cloud), serving all clients that are taking requests from mobile and web clients and relevant IoT observations as input, executes appropriate data processing algorithms through Machine Learning technology, and generates knowledge output by Open APIs; this data analysis is later presented to users.

5. Integration of IoT-generated information through Business Intelligence (BI) analysis into the business processes of an enterprise. This gains importance as a factor in day-to-day business or business strategy definition with the increased use of IoT data by enterprises.

6. The user interface (web or mobile): a visual representation of measurements in a given context (for example on a map and dashboard) and interaction with the user.

It is important to highlight that one of the crucial factors for the success of IoT is stepping away from vertically-oriented, closed systems, and towards open systems, based on open APIs and standardized protocols at various system levels.

Requirements for IoT Architecture

Internet of Things not only links connected electronic devices via the Internet; it is also a web-enabled data exchange that gives systems more capabilities, or “smartness”. In other words, IoT aims to integrate the physical world with the virtual world by using the Internet as the middleware to communicate and exchange information.

There are some specific requirements for IoT that are unconventional to IoT devices and the environments that support them. For instance, many requirements emerge from the limited form-factors and power available to IoT devices. Other requirements come from the way in which IoT devices are manufactured and used; the approaches are much more like traditional consumer product design than existing Internet approaches. Of course, there are a number of existing best practices for the server-side and Internet connectivity that need to be remembered and factored in. We can summarize the overall requirements into some key categories:

- Connectivity and Communications
- Device Management
- Data Collection, Analysis, and Actuation
- Scalability
- Security and Privacy

Connectivity and Communications

Existing protocols such as HTTP have a very important place for many devices. Even an 8-bit controller can create simple GET and POST requests and HTTP provides an important unified (and uniform) connectivity. However, the overhead of HTTP and some other traditional Internet protocols can be an issue for two main reasons. Firstly, the memory size of the program can be an issue on small devices; most MCU devices have less than 16KB memory, which limits the capability for sensor data harvesting. However, the bigger issue is the power consumption requirements. In order to meet these requirements, we need a simple, small and binary protocol. We will look into this in more detail below. We also require the ability to cross firewalls.

In addition, there are devices that connect directly and those that connect via gateways. The devices that connect via a gateway potentially require two protocols: one for sensor devices to connect to the gateway, and another from the gateway to the cloud.

Finally, there is obviously a requirement for our architecture to support transport and protocol bridging: for example, we may wish to offer a binary protocol to the device, but allow an HTTP-based API (RESTful (Representational State Transfer) API) to control the device, which we expose to developers and system integrators.

Device Management

Nowadays, while many IoT devices are not inherently managed, this is certainly not ideal. We have seen active management embedded in PCs, mobile phones, and other devices become increasingly important, and the same trend is likely and desirable for IoT devices. The following list covers some widely desirable requirements:

- The ability to disconnect a rogue or stolen device
- The ability to ren ew the software on a device
- Upgrading security credentials
- Remotely enabling or disabling certain hardware capabilities
- Locating a lost device
- Remotely reconfiguring Gateway, Router parameters

This is not a list of must-have requirements in IoT devices; requirements may be limited depending on device capability.

Data Collection, Analysis, and Actuation

IoT devices are focused on offering one or more sensors, one or more actuators, or a combination of both. The requirements of the system are such that we can collect abundant data from large numbers of devices, and after the data is stored and analyzed, act upon it.

The WISE-IoT is designed to manage numerous devices. These devices create streams of information, which taken together, accumulates a significant amount of data. The requirement is for a highly scalable database system, which can handle diverse data and high volumes, and provide data search results promptly.

The action may happen in near real time, so there is a strong requirement for real-time analytics. In addition, the system also needs to be able to analyze and act on data. In some cases this automatic action will require simple small, embedded logic. On more powerful systems we can utilize more powerful rule engines for event processing and action.

Scalability

The WISE-IoT would ideally be highly scalable, and to be able to support millions of devices all constantly sending, receiving, and acting on data. An important requirement for WISE-IoT is to support scaling from a small deployment to a very large number of devices. Elastic scalability and the ability to deploy in a cloud such as Microsoft® Azure are essential. The ability to scale the WISE-IoT out on an on-premise server is an important requirement for making this an affordable architecture for small deployments as well as large.

Security and Privacy

IoT presents security-related challenges that are identified in the IERC 2010 Strategic Research and Innovation Roadmap but some elaboration are useful as there are further aspects that need to be addressed by the research community. While there are a number of specific security, privacy and trust challenges in IoT, they all share a number of transverse non-functional requirements:

- Lightweight and symmetric solutions, Support for resource constrained devices
- Scalable to billions of devices/transactions Solutions will need to address federation/administrative co-operation
- Heterogeneity and multiplicity of devices and platforms
- Intuitively usable solutions, seamlessly integrated into the real world

Security and privacy issues should be examined very seriously because aside from the fact that IoT handles huge amounts of sensitive data (personal, business data, etc.), it also brings about an influence to the physical environment with its control abilities. Cyber-physical environments must thus be protected from any kind of malicious attacks. The IoT should autonomously tune itself to different levels of security and privacy, while not affecting the quality of service and quality of experience. Security attacks in autonomic and self-aware IoT systems in safety context (e.g. driving cars) can become even more serious because the implementation of a security threat can impact the safety of a user by disrupting the autonomous process.

This concludes the set of requirements that we have identified for the WISE-IoT. Of course, any given architecture may add further requirements. Some of those may already be met by the architecture, and some may require further components to be added. However, our WISE-IoT design is for a modular architecture that supports extensions that cope with this demand.

The Advantech WISE-IoT Architecture

The architecture consists of a set of layers. Each layer performs a clear function. Layers can represent specific technologies, and we will discuss options for implementing each layer. There are also some horizontal/vertical layers such as sensing devices and embedded platforms in the edge field, SaaS (Software as a Service) and SRP (Solution Ready Platform) in the Cloud service, the WISE-PaaS software and foundation as vertical components in this architecture.

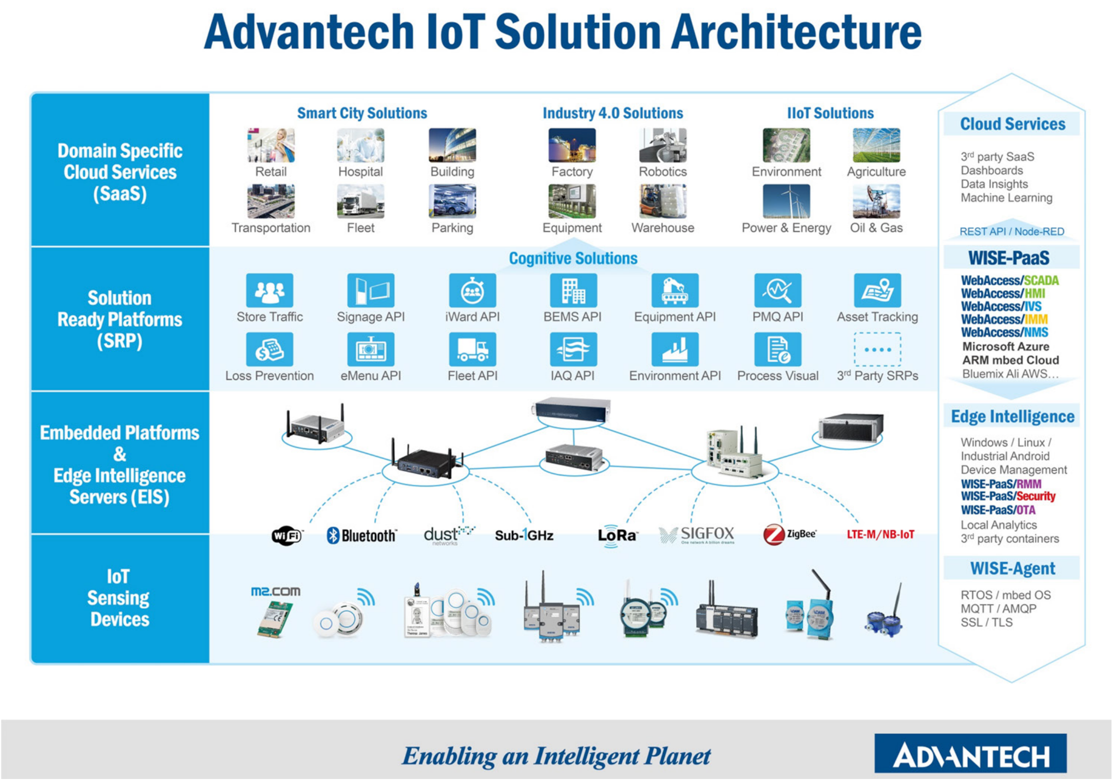


Figure 1. Advantech IoT Cloud Structure

The Layers are:

- IoT Sensing Devices
- Edge Intelligence Servers, Embedded Platforms (EIS)
- Solution Ready Platforms (SRP)
- Domain Specific Cloud Services (SaaS)

The horizontal layers are:

- WISE-Agent
- Edge Intelligence
- WISE-PaaS
- Cloud Services

The IoT Sensing Devices Layer

The bottom layer of the architecture is the device layer. A device can be one of a number of different types, but in order to be considered an IoT device, it must have some communications that either indirectly or directly attach to the Internet. Examples of direct connections are:

- Embedded systems with Ethernet connection.
- Automation controller with Ethernet or Wi-Fi connection.
- Automation controller with RS485/RS232 or Ethercat connection.
- WISE-IoT devices connect via Ethernet or Wi-Fi connection.
- Embedded systems with the wired interface of I^{2}C, SMBus, USB, PCIExpress, and so on.

Examples of indirectly connected devices include:

- 6LoWPAN devices connect via a 6LoWPAN gateway.
- SubG devices connect via a SubG gateway.
- RFID devices connect via a RFID receiver.

Most indirect connection protocols request low power communication; several low power communication technologies have been proposed from different standardization bodies. The most common ones are:

- IEEE 802.15.4 (6LoWPAN, ZigBee, SubG) has developed a low-cost, low-power consumption, low complexity, low to medium range communication standard at the link and the physical layers for resource constrained devices. The WISE-IoT 6LoWPAN solution has adapted the Linear Technology SmartMesh WSN solution.

- Bluetooth Low Energy (Bluetooth LE) is the ultra-low power version of the Bluetooth technology that is up to 15 times more efficient than Bluetooth.
- Ultra-Wide Bandwidth (UWB) Technology is an emerging technology in the IoT domain that transmits signals across a much larger frequency range than conventional systems. UWB, in addition to its communication capabilities, can allow for high precision ranging of devices in IoT applications.
- RFID/NFC proposes a variety of standards to offer contactless solutions. Proximity cards can only be read at a distance of less than 10 cm and follow the ISO 14443 standard, which is also the basis of the NFC standard. RFID tags or vicinity tags dedicated to the identification of objects have a reading distance which can reach 7 to 8 meters.

The current M2M (Machine to Machine) related standards and technologies landscape is highly fragmented. The fragmentation can be seen across different verticals and application specific domains where there is very little or no leverage of technologies beyond basic communications or networking standards. Even within a particular application sector, a number of competing standards and technologies are used. The entire ecosystem of solution providers and users would greatly benefit from less fragmentation and should strive towards the use of a common set of tools. This would provide faster time to market, economy of scale and reduce overall costs.

In the future, the number and types of IoT devices will increase dramatically; therefore interoperability between devices will be essential. More computation and yet less power and lower cost requirements will be essential.

The standardization bodies are addressing the issue of interoperable protocol stacks and open standards for the IoT. This also includes expanding the HTTP, TCP/IP stack to the IoT-specific protocol stack. This is quite challenging considering the different wireless protocols like ZigBee, RFID, Bluetooth, BACnet 802.15.4e, 6LoWPAN, RPL, and CoAP.

To drive further standardization of device technologies in the direction of standard internet protocols and web technologies, and towards the application level, would reduce the impacts of fragmentation and strive towards true interoperability. Embedded web services, as driven by the IETF (Internet Engineering Task Force) and IPSO (Internet Protocol for Smart Objects) Alliance, will ensure a seamless integration of IoT devices with the internet. It will also need to include well-formed representation of IoT device-hosted services and capabilities.

The bottom layer of devices consists of Advantech platforms with the capacity to connect to the Internet either indirectly or directly. The WISE-devices are typically System-on-Chip, such as ARM cortex-M series integrated low power consumption protocols, for example, 6LoWPAN, SubG and IBM Lora, the objective is providing long battery life and long-period data observations. The WISE-Devices are usually resource limited and typically have no operating system or run embedded Linux platforms such as OpenWRT, or dedicated real-time embedded operating system such as FreeRTOS or mbed OS. The interoperability and communication between devices is compatible with IETF and IPSO alliance, and also compatible to CoAP in communications.

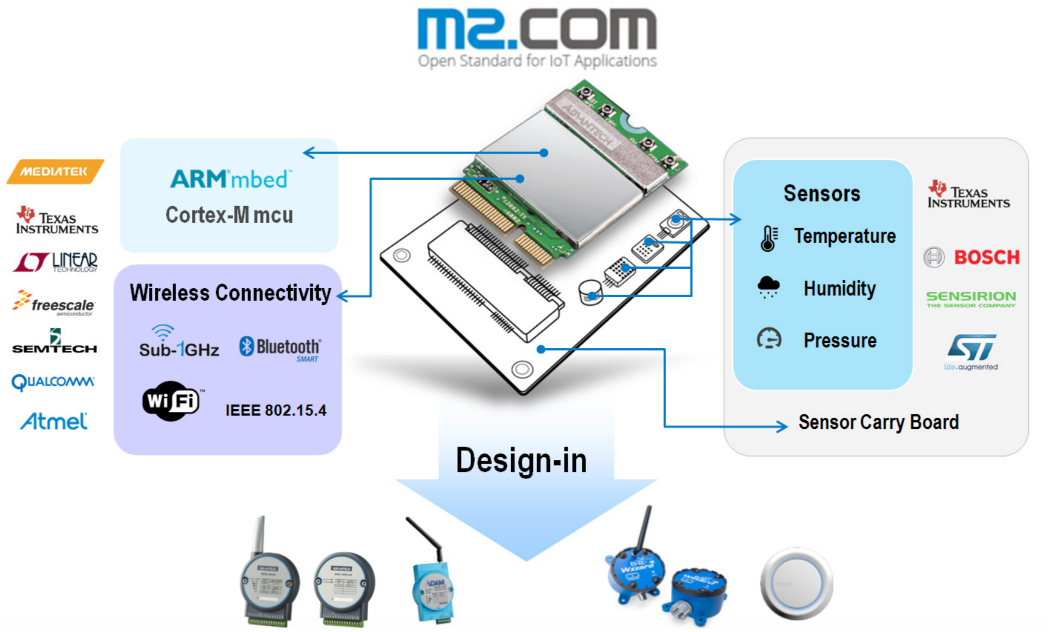


Figure 2. Advantech M2.COM standard

The Internet of Things (IoT) will remain fragmented until at least 2018, with no dominant ecosystem, providers or technical model to set industry standards, according to Gartner.

Gartner research director Alfonso Velosa said in the analyst firm's Predict 2015: The Internet of Things report that IT leaders will need to find ways to make use of the IoT from multiple providers and sources that are future proof.

Advantech has announced the M2.COM open standard, and we believe this is a step towards reducing IoT fragmentation as Figure 2 shows.

The M2.COM standard is not only unified HW interface leverage M.2 standard but also unified the operating system of ARM provided mbed OS for Cortex-M MCU.

We partner with ARM and RF chip vendors such as MediaTek, TI, Linear Tech, and Qualcomm to provide different kinds of RF technologies such as Wi-Fi, 6LowPAN, and Sub-1G,LoRa, Sigfox, etc., with the goal to place the MCU + RF together on top of the module board as per the M2.COM form factor definition. We also partner with sensor chip vendors such as Bosch, TI, and ST who has domain specific sensing knowledge to design their application specific board on bottom as a carrier board. Advantech has application product teams following this standard to create housings for different usages in critical environments. For example, rugged IP65, waterproof enclosure fully enclosed with the M2.COM module sensor board and battery.

The Edge Intelligence Servers, Embedded Platforms (EIS) Layer

The EIS layer supports the connectivity of the devices. There are multiple potential protocols for communication between the devices and the cloud. The most well-known four potential protocols are:

- HTTP/HTTPS (and RESTful approaches on those)
- MQTT 3.1 / 3.1.1
- AMQP
- Constrained Application Protocol (CoAP)

Let’s take a quick look at each of these protocols in turn.

HTTP is well known, and there are many libraries that support it. Because it is a simple, text-based protocol, many small devices such as 8-bit controllers can partially support the protocol. For example, often even limited devices can support enough code to POST or GET a resource. The larger 32-bit based devices can utilize full HTTP client libraries that properly implement the whole protocol.

In addition to HTTP, there are several protocols optimized for IoT use. The two best known are MQTT (MQ Telemetry Transport)and CoAP (Constrained Application Protocol). MQTT was invented in 1999 to solve issues in embedded systems and SCADA. It has been through some iterations and the current version (3.1.1) has now become standardized by the OASIS MQTT Technical Committee. MQTT is a publish-subscribe messaging system based on a broker model. The protocol has a very small overhead (as little as 2 bytes per message) and was designed to support lossy and intermittently connected networks. MQTT was designed to flow over TCP. In addition, there is an associated specification designed for Sensor Networks called MQTT-SN that is aimed at embedded devices on non-TCP/IP networks, whereas MQTT itself explicitly expects a TCP/IP stack, such as ZigBee.

The Advanced Message Queuing Protocol (AMQP) is an open standard application layer protocol for message-oriented middleware. The defining features of AMQP are message orientation, queuing, routing (including point-to-point and publish-and-subscribe), reliability and security. This protocol is more and more popular and reorganized as an OASIS member section in August 2011. Microsoft has announced that the Azure service bus supports AMQP 1.0, and WISE-PaaS is going to take the next step to be our standard alternate protocol.

The Constrained Application Protocol (CoAP) is a protocol from the IETF that is designed to provide a RESTful application protocol modeled on HTTP semantics, but with a much smaller footprint and a binary rather than text-based approach. CoAP is a more traditional client-server approach rather than a brokered approach. One difference between HTTP and CoAP is the transport layer. HTTP relies on the Transmission Control Protocol (TCP). TCP's flow control mechanism is not appropriate for LLNs (Low power and Lossy Networks) and its overhead is considered too high for short-lived transactions. In addition, TCP does not have multicast support and is rather sensitive to mobility. CoAP is built on top of the User Datagram Protocol (UDP) and therefore has significantly lower overhead and multicast support. However, since the CoAP is the ARM mbed OS standard protocol, the WISE-IoT also supports this protocol for field side sensor devices with transit to MQTT protocol on the Gateway, and then passes to WISE-PaaS as well.

For the WISE-IoT architecture, we have opted for MQTT as the preferred device communication protocol, with HTTP as an alternative option. The reasons for selecting MQTT and not CoAP at this stage are:

- Better adoption and wider library support for MQTT; there are 24 different programming language libraries provided on the official web site.
- Simplified bridging into existing event collection and event processing systems.
- Simpler connectivity over firewalls and NAT networks.

However, both protocols have specific strengths (and weaknesses) and so there will be some situations where CoAP may be preferable and could be swapped in.

In order to support MQTT, we need to have an MQTT broker in the architecture, as well as device libraries. We will discuss this with regard to security and scalability later.

One important aspect with IoT devices is not just for the device to send data to the cloud/ server, but also the reverse. This is one of the benefits of the MQTT specification: because it is a brokered model, clients make an outbound connection to the broker, whether or not the device is acting as a publisher or subscriber. This usually avoids firewall problems, because this approach works even behind firewalls or via NAT (Network Address Translation).

http://mqtt.org

http://tools.ietf.org/html/draft-ietf-core-coap-18

http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html

In the case where the main communication is based on HTTP, the traditional approach for sending data to the device would be to use HTTP polling. This is very inefficient and costly, both in terms of network traffic as well as power requirements. The modern replacement for this is the WebSocket protocol, which allows an HTTP connection to be upgraded to a full two-way connection. This then acts as a socket channel (similar to a pure TCP channel) between the server and client. Once that has been established, it is up to the system to choose an ongoing protocol to tunnel over the connection.

The Solution Ready Platform (SRP) Layer

This is an application layer to fulfill the diverse demands for specific domain applications; the main service of the API helps IoT customers to easily integrate their existing solutions into IoT applications. This layer can include three kinds of components:

1. Domain specific HW and sensor solutions such as cameras and data server for video surveillance.
2. The foundation of WISE-PaaS for Cloud connection and communication.

3. 3rd party solutions to accomplish real domain applications such as integrating an algorithm into a camera so that it can pattern-match video data for further analytics.

Advantech invites more and more 3rd party solutions to enrich portfolios on the IoT SRP solutions layer in order to fulfill numerous diverse IoT requirements from different vertical domains. The open standards framework is a basic platform that carries many such solutions on the cloud at once.

The Domain Specific Cloud Services Layer

The WISE-IoT Architecture needs to provide a way for these devices to communicate outside of the device-oriented system. This includes three main approaches.

- The ability to create web-based front-ends and portals that interact with devices and with the event-processing layer.
- The ability to create dashboards that offer views into the analytics and event processing.
- The ability to interact with systems outside this network using machine-to-machine communications (APIs).

These APIs need to be managed and controlled, and this happens in a RESTful API Management system.

The recommended approach to building the web front end is to utilize a modular front-end architecture such as a portal, which allows simple, fast composition of useful UIs. Of course, the architecture also supports existing Web server-side technology such as Java Servlets/JSP, PHP, Python, etc. Our recommended approach is based on the Java framework and the most popular Java-based web server: Apache Tomcat.

The dashboard is a reusable system focused on creating graphs and other visualizations of data coming from the devices and the event processing layer. These visualizations of analyzed data, often enhanced by Machine Learning, are more valuable for enterprise adaptation than mere raw data. The RESTful API Management layer provides three main functions:

- It provides a developer-focused portal, where developers can find, explore and subscribe to APIs from the system; there is also support for publishers to create, version and manage the available and published APIs;
- The second is a gateway that manages access to the APIs, performing access control checks (for external requests) as well as throttling usage based on policies. It also performs routing and load-balancing;
- The final aspect is that the gateway publishes data into the analytics layer where it is stored and processed to provide insight into how the APIs are used.

The Horizontal Software Layer

This layer is associated with software components that interact with nearby vertical layers; there are more explicit descriptions of the detailed technologies for each of the layers in the next section.

- The WISE-Agent follows the MQTT standard protocol and the source code of cross-platform programming C language goes on the GitHub server for operating systems such as Windows, Linux, RTOS, etc. that are handling the communications and handshakes between layers.
- The Edge Intelligence is based on the container technology framework, which connects the southbound and northbound and intelligent technology modules, such as data preprocessing and lightweight analytics.

- The WISE-PaaS is an important layer of the architecture; it aggregates and brokers communications. It is important because it provides these three abilities:

  1. The ability to permit an HTTP server and an MQTT broker to talk to the devices
  2. The ability to aggregate and combine communications from different devices and to route communications to a specific device (possibly via a gateway)
  3. The ability to bridge between and transform different protocols. For example, HTTP-based APIs (RESTful APIs) can be transferred into an MQTT message going to the device.

  The WISE-PaaS layer provides these capabilities as well as adapting to legacy protocols. The WISE-PaaS layer may also provide some simple correlation and mapping from different correlation models (e.g., mapping a device ID into an owner's ID or vice-versa).

  Advantech has developed various software applications for specific domain applications with different foundation management needs.

- The Cloud Infrastructure

  This is part of the WISE-PaaS Cloud infrastructure. WISE-PaaS is allied with Microsoft Azure IaaS (Infrastructure as a Service) and implements Azure PaaS modules to enhance WISE-PaaS functionality. There are two main forms:

  1. Virtual machine, for flexible deployment of different IaaS providers
  2. Cloud Clustering, for in/out scalability to cope with IoT demand.

  WISE-PaaS can also be implemented on some IaaS providers such as Amazon AWS, IBM blueMix, and Baidu, etc. that is based on the cloud foundry to build up the WISE-PaaS framework. For this adoption to be more flexible to deploy WISE-PaaS to On-Premise or Cloud-Base infrastructure, the WISE-PaaS supports three kinds of modes, which include the following:

  1. Deploy the WISE-PaaS to a Public Cloud
  2. Hybrid mode for private cloud and public cloud
  3. Private Cloud through virtual private network (VPN) connection

- The WISE-PaaS/RMM

  The WISE-PaaS/RMM (Remote Management and Monitoring) is a combination of two major components. A server side system (the Device Manager, or DM) communicates with the devices via various protocols and provides both individual and bulk control of devices. It also remotely manages the software and applications deployed on devices. It can monitor and/or reset device applications or processes if necessary. The Device Manager works in conjunction with the device management agent (WISE-Agent). There are multiple different agents for different platforms and device types. The Device Manager also needs to maintain the list of device identities and map these to owners. It must also work with the Identity and Account Management block to manage access controls over devices (e.g., who else can manage the device apart from the owner, how much control does the owner have vs. the administrator, etc.).

- WebAccess+

  The WebAccess+ are including four vertical markets specific applications:

  1. WebAccess/SCADA: traditional SCADA system for automation applications, such as PLC devices.
  2. WebAccess+IVS: Intelligent video surveillance for video analytic based applications.
  3. WebAccess+IMM: Interaction multimedia for HMI and signage applications
  4. WebAccess+NMS: Network management systems

- The 3rd Party Solutions

  The IoT Cloud will not entirely be developed by Advantech alone; we have also invited many Cloud providers such as ARM mbed, IBM®, Intel® and Cisco to join this partnership, all helping adapt each solution to fulfill IoT demands across diverse applications. The collaboration model should be available through the Open APIs provide by each Cloud provider. Communication via both cloud and SaaS layer can benefit domain applications and system integrators.

- The Security

  The WISE-PaaS provides three layers of security:

  1. Operating System: McAfee
  2. Communications: OpenSSL (Transport Layer Security), TLS (Secure Sockets Layer), DTLS (Datagram Transport Layer Security)
  3. Internet: Front-end Web Security

For operating system malware attack prevention, Intel® McAfee and the white list mechanism feature hardens the embedded operating system to prevent any non-listed applications or processes from executing. For communication, we are compatible with SSL embedded communication protocols between the server and client sides; we have also implemented TLS and DTLS data encryption. For web server flaw detection, WISE-PaaS has been certified by market leading software such as OpenVAS and Nessus.

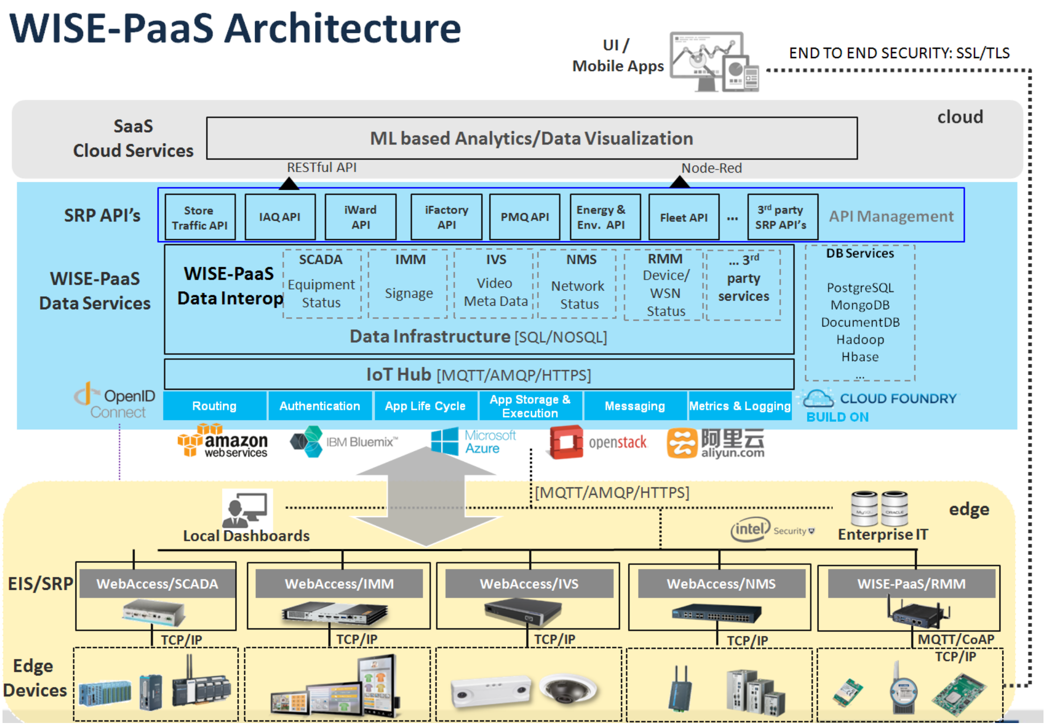


Figure3. The WISE-PaaS Open Architecture

WISE-PaaS Open Architecture

In order to meet Advantech IoT architecture requirements, software efforts should always cover all devices such as MCU, SOC, Atom™, x86 and ARM architecture systems. As shown in Figure 3, WISE-PaaS open architecture implements three kinds of software building blocks. They include:

- A software solution for domain applications and embedded platforms on the edge
- WISE-PaaS Open architecture that translates data from edge solutions and 3rd party solutions, based on the open standard framework
- RESTful API and Node-RED tools that help integrate a solution with built-in analytic capability

At the bottom of this open architecture, Advantech has five kinds of edge IoT solutions for specific applications:

- The WebAccess/SCADA system that deals with traditional protocols and equipment data transformation such as Modbus, Profibus, OPC UA, etc. This software has a lot of traditional drivers such as PLC to transform the driver layer to a data layer, and furthermore to transform such data to IoT standard formats such as IETF and CoAP as defined in MQTT/AMQP protocols.

- The WebAccess/IMM system, a digital signage solution with dashboard options for end users and administrators. Data from user interactions can often be valuable information for vertical applications.

- WebAccess/IVS system, a video solution where camera data is transformed into meaningful information such as age, gender, emotion, etc. The data is also transfered to WISE-PaaS for further analytics.

- WebAccess/NMS system, IoT management through SNMP protocol tracks network topology and health. This is essential data for any IoT solution.

- WISE-PaaS/RMM system, the backbone of WISE-PaaS, mainly handles communication and connectivity tasks for embedded devices, including device management, health, and remote control through MQTT protocol. Facilitates customer development of their own applications, based on open SDK and APIs.

These five existing Advantech solutions conduct data unification and transfer it to a WISE-PaaS database for further utilization such as real-time dashboard for user interaction or analytic dashboard tools geared toward ends such as operation optimization or equipment predictive maintenance. Sensitive data can be stored in a secure, local database, with a local dashboard providing real-time interaction between humans and equipment; real-time response can be vital in case of critical events.

The middle layer is our WISE-PaaS; Advantech is leveraging a couple of open technologies to build in flexibility and scalability that can service demands for numerous connections in the future. There are seven aspects to the WISE-PaaS architecture.

- The clustering IoT hub for connectivity, in order to deal with burst connections from edge sensing devices, the cluster connection design is a crucial component in WISE-PaaS architecture with the flexibility to fulfill diverse cloud position requirements (private, hybrid, public). Furthermore, the connection component needs to be exchangeable with a commercial IoT hub, for example, Azure IoT hub and Ali IoT hub also.

- The WISE-PaaS data infrastructure is like twin Advantech IoT engine solutions in the edge. Each data infra block is dedicated to handling data flow for the corresponding solution; the block should have two main flows, one to the database and one to the frontend for real-time interaction. The data interoperation of all of WISE-PaaS blocks facilitates SRP API to design IoT applications with Advantech key IoT solutions at the same time.

- The Cloud Foundry PaaS framework is an open source, multi-cloud application platform as a service (PaaS), governed by the Cloud Foundry Foundation organization. Advantech leverages this PaaS framework for load-balancing, software deployment and lifecycle management, container function, etc.; this framework has been used in GE's Predix, IBM Bluemix, Pivotal cloud foundry, and so on.

- The Single Sign-On (SSO) is composed of OAuth 2.0 and OpenID. OAuth 2.0 focuses on client developer simplicity while providing specific authorization flows for web applications, desktop applications, mobile phones, and living room devices. This specification is being developed within the IETF OAuth WG, and is based on the OAuth WRAP (Web Resource Authorization Protocol) proposal. OpenID is an open-standard, decentralized authentication protocol. There are a lot of organizations that have adopted OpenID on their websites already, such as Google, AOL, Amazon, Microsoft, BBC, IBM, etc. Advantech WISE-PaaS adopted SSO to authorize further access to edge IoT solutions and provided a single sign-on web portal on WISE-PaaS to simplify heterogeneous systems integration.

- The SRP API and API management is a RESTful API layer comprised of WISE-PaaS RESTful API's set of accommodated Advantech edge IoT solutions and 3rd party IoT solutions that also follow the open architecture. The APIs involved would be the specific definition of those utilized in the vertical domains, such as energy & environment API, Intelligent Air Quality (IAQ) API, and so on and so forth. The API management would be the next integration for the APIs, combining tooling system and API usage counting management.

- The SaaS cloud service conducts the machine learning and data analytics systems, with domain expertise and data engineers applied to working out the insight values of your particular IoT application; machine learning results could be implemented either in the cloud or in an edge device, depending on computing resources and efficiency considerations.

- The UI Mobile App is the latest human interaction device that would run on a mobile device such as a smart phone, tablet, or notebook. The newest mobile devices support HTML5, the latest WWW markup language; more and more industrial devices have adopted mobile devices for presentation and interaction purposes. The RESTful API is also designed to use HTML5 to fulfill complex web applications.

Advantech WISE-PaaS would like to provide WISE-PaaS as the real IoT sharing platform for heterogeneous systems; Advantech WISE-PaaS uses this architecture to prove this concept for integration, and invites 3rd party SW vendors to adopt the same framework to offer their solutions as well.

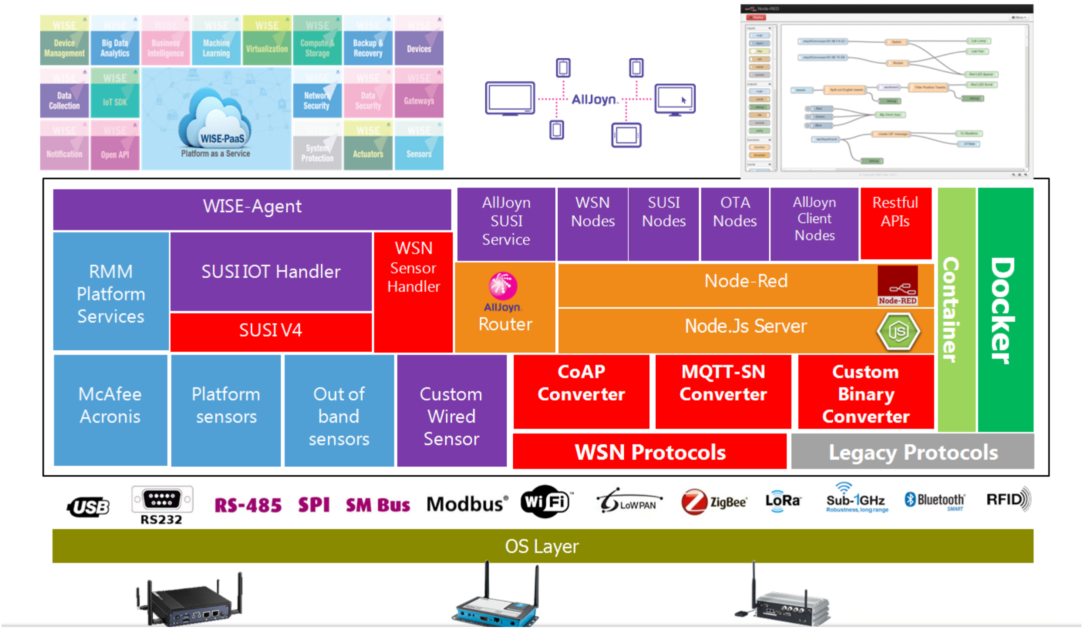


Figure 4. Edge Intelligence Software Block Diagrams

The Advantech Edge Intelligence Software Diagrams

Edge Intelligence is a modular design for flexible integration of any kind of module. As presented in Figure 4, it has implemented basic WISE-PaaS/RMM features such as device management, energy saving, KVM, system protection, backup/recovery, and so on; it also provides standard APIs for the in-band sensor, the same ones that usually use SUSI API for platform sensors such as fan speed, CPU/SYSTEM temperature, specific voltage, etc. Moreover, expanded APIs are also available for out-of-band sensors produced by Advantech such as SAB-2000, POE, SQFlash, m2talk, and so on, as peripherals devices.

In the Advantech Edge Intelligence software stack as shown in Figure 4, there are 3 key benefits:

- Cloud Connection: Advantech Edge Intelligence Server has WISE-Agent and Microsoft Azure IoT suite embedded. A few simple steps connect this solution to the IoT Cloud.

- Cross Interoperability: The Edge Intelligence Server is integrated with AllJoyn and IoTivity IoT standard libraries, and also offers Advantech AllJoyn gateway service to interact with any AllJoyn compliant devices.
- Local Intelligence: The gateway solution is integrated with the Node-RED design tool, and also offers Advantech add-on nodes such as SUSI API nodes, WSN API nodes, WISE-PaaS nodes, and AllJoyn nodes to help the customer quickly design their application's logic using drag and drop.

Advantech also provides a lot of add-on values to help facilitate customer adoption of IoT application sensors using our APIs, and our graphics based development environment such as Node-RED, which IBM contributed to the open source community for IoT data flow and application logic design, as shown in Figure 5. Advantech based this environment on node. js for development of a lot of nodes; these take advantage of useful APIs for Advantech devices and sensors. A customer can use those nodes to design interaction between different applications. For example, a customer can use WSN API to get the remote environment temperature and set a well threshold to monitor, and interact with the device's PWM. This would enable the customer to control its fan switch and speed with a wired interface.

In addition, we integrate the AllJoyn standard established by Microsoft and Qualcomm; this standard helps with interoperability between heterogeneous systems at the same layer. As shown in Figure 4, in sensor date format, they are all compatible with IPSO alliance (SENML) in every function block of Edge Intelligence.

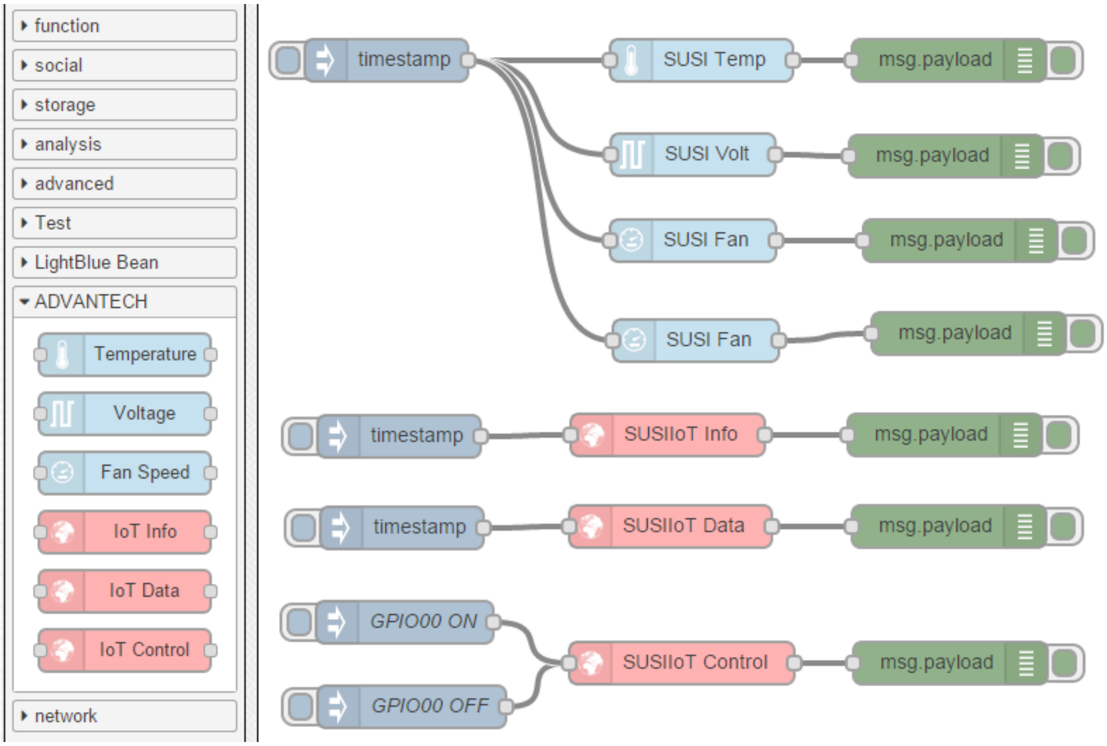


Figure 5. Advantech Node-RED Nodes

Advantech has put these add-on nodes on the GitHub for easy customer access. You can download and try out these items on the Advantech platform. Here is the related link https://github.com/ADVANTECH-Corp

We also put detailed information on the Advantech technical wiki for your reference. You can find the detailed technical information and SOP for usage in Node-RED.

http://ess-wiki.advantech.com.tw/view/SW_Service/Node-Red_for_SW_Service

http://ess-wiki.advantech.com.tw/view/IoTGateway/Node-Red

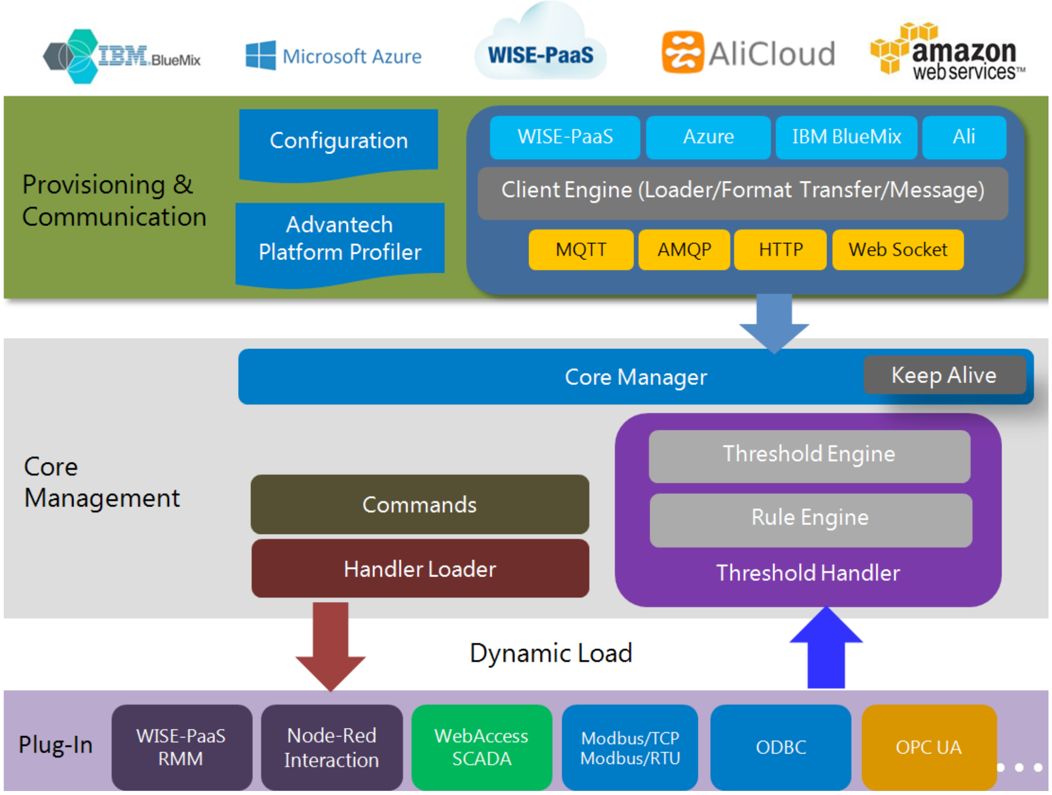


Figure 6. WISE-Agent Architecture

Let's take a look at the WISE-Agent architecture in more detail. As seen in Figure 6, the WISE-Agent has a dynamic loading function handler (plug-in) library and opens the 3rd interface for an extra function handler for developer programming and integration with their application. In addition, the WISE-Agent can load different protocols for connection of existing gateways or edge computer solutions. This modular design means more flexible load transmission protocols to MQTT to link to WISE-PaaS. This modularization provides flexibility and scalability advantages for quick adaptation from any existing M2M solution.

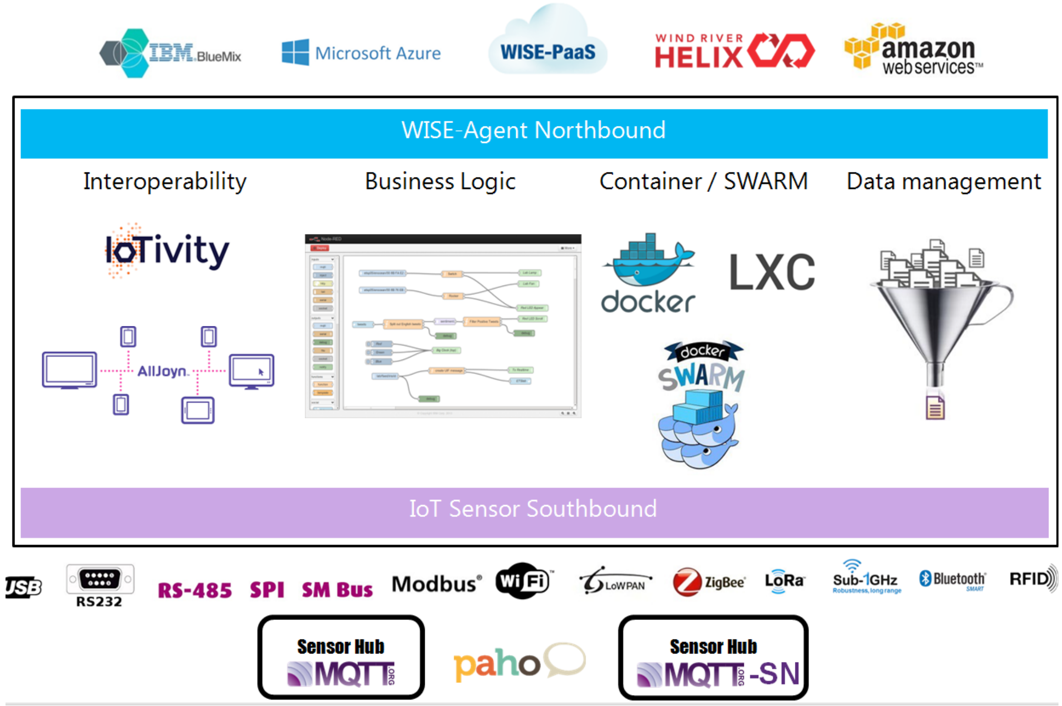


Figure 7. Edge Intelligence: Three Major Functionalities

The Edge Intelligence Server decouples three major functionalities; i.e., Northbound, Edge Intelligence, and Southbound data flows. From the edge device point of view, the edge device should have the capability to connect to multiple cloud solutions, and transmit different cloud protocols such as MQTT, AMQP and HTTP. It should be easy to tailor integration of any resource constricted operating system (e.g. FreeRTOS) by modularized northbound design as shown in Figure 7.

As mentioned, for multiple sensor protocol conversion as found in IoT southbound technology, framework design must deal with different sensors' protocol transformation and transmission, including wired devices and wireless devices with IP and non-IP protocols. Advantech Edge Intelligence server integrates MQTT Broker as the central manager to handle the northbound connection, intelligence and interaction with cross-gateway systems for heterogeneous IoT systems.

Edge intelligence involves interoperability, business logic, container, SWARM, and data management technologies. As Figure 7 shows, Advantech has integrated AllJoyn and IoTivity for interoperability functionality, and has developed a number of Node-RED service nodes with local business logic to reduce the design effort required to create IoT solutions. We are implementing container technology that promotes easy solution deployment, and leveraging different operating systems. Our plan is to implement data management for pre-processing of data in motion and data filtering in side sensor data collection. SWARM technology is the hottest topic in the IoT, and is being widely implemented in the field. Advantech is still tracking this technology and collaborating on academic research regarding related topics.

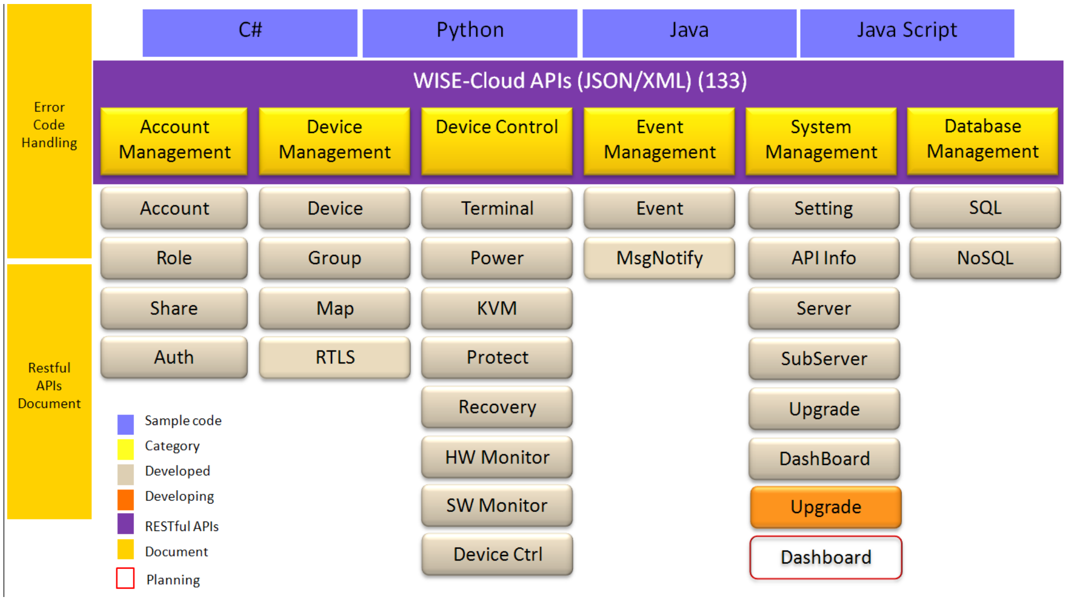


Figure 8. WISE-PaaS RESTful APIs Category

Based on the IEFT definition, the RESTful API follows certain design guides; the WISE-PaaS RESTful APIs are compatible with those guidelines, which makes them easy for web developers to understand and to develop applications quickly. As presented in Figure 8, the WISE-PaaS RESTful APIs provide fundamental function categories (see yellow blocks). WISE-PaaS for the IoT sensor category was released Q1 2016, and provides valuable RESTful APIs to access dedicated databases for business intelligence, and widget APIs for dashboards that show specific analytic results.

[RFC4627]: RFC 4627 - The application/JSON Media Type for JavaScript Object Notation (JSON)

[RFC6570]: RFC 6570 - URI Template

[RFC4288]: RFC 4288 - Media Type Specifications and Registration Procedures

[RFC2396]: RFC 2396 - Uniform Resource Identifiers (URI): Generic Syntax

[RFC2616]: RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1

IoT Interoperability

The Internet of Things is shaping the evolution of the Internet of the future. After connecting people all the time and everywhere, the next step is to interconnect heterogeneous things/machines/smart objects both between themselves and with the Internet, and allowing the creation of value-added open and interoperable services/applications, enabled by their interconnection, in such a way that they can be integrated with current and new business and development processes. There are four technical IoT interoperability aspects:

1. **Technical Interoperability:** The WISE-PaaS is usually associated with hardware/software components, systems, and platforms that enable machine to machine communication to take place. This kind of interoperability relies mainly on communication protocols and the WISE-PaaS needed for those protocols to operate. The WISE-PaaS uses MQTT as the main communication protocol for now. The flexible WISE-Agent design will help implement multi-protocols in the future as well.

2. **Syntactical Interoperability:** The WISE-PaaS is usually associated with data formats. Certainly, the messages transferred by communication protocols need to have a well-defined syntax and encoding, even if it is only in the form of bit-tables. The WISE-PaaS uses XML and JSON formats, which can be represented using high-level transfer syntaxes.

3. **Semantic Interoperability:** The WISE-PaaS is usually associated with the semantic comprehension of the content understood by human rather than a machine. Thus, interoperability on this level means that there is a common understanding between people on the content being exchanged (interpret variables that represent complex symbolic operations), as opposed to merely decoding words. The WISE-PaaS is compatible with IEFT SENML (Media Types for Sensor Markup Language) definition, and the AllSeen alliance defined AllJoyn which Microsoft Windows 10 has been supported by default.

4. **Organizational Interoperability:** as the name implies, WISE-PaaS RESTful API is the ability of organizations to effectively communicate and transfer meaningful data or information, even though they may be using a variety of different information systems over widely different infrastructures, and possibly across different geographic regions and cultures.

Conclusion

The WISE-PaaS objective is that through the collaboration and integration of multiple vertical market data sources, it can reach more powerful solutions, get better data-analysis, create more accurate data-driven modeling and situational awareness plus make better, more definitive solutions. For example, with regard to green energy and building automation, we hope that WISE-PaaS can continue to offer ever increasing energy optimization with associated cost reductions for all of us.

The Internet of Things vision is evolving rapidly, and of course there are associated technological challenges. Regardless of these challenges, the WISE-PaaS architecture is based on real world projects that we have deployed with customers to support IoT capabilities. We have great confidence that this is a scalable, useful, deployable, and effective architecture.

Advantech has been a world-leader in providing trusted, innovative embedded and automated products and solutions since 1983. As we enter the IoT era, Advantech has expanded into providing comprehensive solution ready platforms incorporating both hardware and software layers to enable customers to develop their own IoT solutions without apprehensions.

Works Cited:

[1] Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems.

http://www.internet-of-things-research.eu/pdf/Converging_Technologies_for_Smart_Environments_and_Integrated_Ecosystems_IERC_Book_Open_Access_2013.pdf

[2] A reference architecture for the internet of things

https://www.researchgate.net/profile/Paul_Fremantle/publication/308647314_A_Reference_Architecture_for_the_Internet_of_Things/links/57ea00b708aef8bfcc963153.pdf