IoT Software Development

IoT deployments can offer a variety of use cases across all industries, but engineers must design their applications with the right platforms, OSes and programming languages. IoT devices give organizations the capability to automate various business operations, including everyday activities.

Device integration can include connecting mobile phones, TV, smartwatches, laptops and sensors -- such as accelerometers or gyroscopes -- with one another. IoT infrastructure and devices must communicate efficiently to make the devices operational for business and consumer use cases. This is where IoT-oriented software engineering comes into play.

IoT software engineering refers to the systematic development of IoT infrastructure using a blend of software and hardware solutions. IoT software engineering deals with data collected via sensors and processes it to make sense for real-world applications with intuitive UI and visual representation.

IoT has applications in many fields, such as aviation and avionics for status checking and sustainable activities, the automobile industry for engine parameters and vehicle safety, broadcast communications or even entertainment. Some of the growing popular applications include: smart cities construction, smart power grids, medical services, smart market-based analysis.

For developing IoT software solutions, organizations require three components: the development platform, OS and programming language. Building and launching an IoT-based product starts with a platform. Developers must consider different characteristics offered by each platform to find their ideal fit. Here are a few examples:

IBM Watson. Tech giant IBM offers Watson as a platform for the development of IoT solutions. It provides quick and secure implementation, online data analysis and critical risk visualization.

Azure. Microsoft offers the Azure IoT development platform with capabilities such as data collection, analysis and visualization. Developers can also scale the IoT application and increase the interoperability between more devices without making vast changes.

AWS. The benefits of choosing AWS IoT include AI integration, multilayered security and scalability. This development platform also offers device software, connectivity and control, and analytics services.

Generally, IoT systems consist of low-powered processing units, small RAMs and restricted storage. These limitations mean the OS needs to be lightweight and not resource hungry. The following are some of the available OSes:

Raspbian. A widely popular OS for IoT systems, Raspbian is optimized for Raspberry Pi hardware. The OS is easy to use and compiles more than 35,000 packages.

Arm Mbed OS. This OS is an open source OS that meets all the requirements for IoT systems. Mbed OS features multilayer security as well as drivers for Bluetooth connectivity, thread, 6LoWPAN, Ethernet and Wi-Fi.

Picking the right programming language is also crucial for developing IoT applications. With limited resources on hand, the code needs to be short and easy to compile. Here's a list of IoT-based programming languages:

C and C++. Both widely known languages are written with a hardware perspective in mind.

Java. Known as the mobile programming language, Java is compatible with various peripheral devices and is well suited for IoT devices.

Python. Python offers a good fit for IoT applications because it can handle data-heavy applications.

When compared to any traditional software development process, the process for IoT architecture is considerably different. An IoT-based system comprises four stages of architecture.

Stage 1: Sensors and actuators

In this stage, the architecture consists of sensors and actuators that collect data from the environment or object and turn it into useful data.

Stage 2: Data preprocessing

The data collected by sensors in stage 1 are usually analog and raw. For IoT applications to make use of this data, it needs to be aggregated and converted into digital data. Data acquisition systems connected to sensor networks perform the analog-to-digital conversion.

Stage 3: IT systems

The converted data can't simply pass into the data center. Engineers must address issues such as lack of space and security. The data must be preprocessed, and only the significant results pass to the cloud.

Stage 4: The cloud

After significant data is extracted, it is stored in either traditional data centers or the cloud. From there, IT professionals can reshape the data and make it readable for the end users.

IoT involves a tremendous exchange of data. Developers and IT teams must consider the data's security and continuously take measures to improve it. The connections between multiple devices to a network expose them to potential security threats. The communication channel is the core of any IoT software engineering. To safeguard it from problems, developers must follow strict protocols and established schemas.

The IoT market is growing rapidly and expected to reach a capital of $561 billion by 2022, according to the MarketsandMarkets Research global forecast. With the growth of IoT-based environments, there will be a rise in the demand for IoT-oriented software engineering.