Platform Architecture Guide:

INTERNET OF THINGS

Application Development

anypresence

anypresence

Table of Contents

anypresence

Platform Architecture Guide: **IoT Application Development**

Introduction

> Spending on IoT will reach $1.7 Trillion by 2020.
>
> - IDC

The Internet of Things represents a massive growth opportunity for businesses across virtually all industry segments. According to IDC, the spending on IoT will reach $1.7 Trillion by 2020 with a compound annual growth rate (CAGR) of 16.9%. Gartner predicts that Smart Homes will lead the IoT momentum with 294 million connected things in use in 2015. Developers will be at the center of this opportunity, helping to craft IoT solutions across both industrial and consumer use cases.

As application developers transition from a mobile device to a broader connected device world, they need a robust and scalable platform that enables them to leverage their existing skills, knowledge and expertise. In recent years, mobile and web developers took advantage of cloud services through mobile backend as a service (MBaaS) platforms, which offered them the essential building blocks of application development. Many components of MBaaS are also relevant and applicable to IoT solution development. The same infrastructure can be used to build the next generation of IoT solutions that go beyond the traditional mobile-first applications, to enable both internal and external IoT app innovation.

This white paper explores a reference architecture for IoT solutions, with an example IoT app platform implementation based on the AnyPresence solution capabilities.

The Building Blocks of an Internet of Things App Platform

During the last few years, mobile developers started to exploit cloud capabilities through a niche service model that is often referred to as Mobile Backend as a Service (MBaaS), or more generically Mobile Infrastructure Services. This new delivery model empowered developers by enabling them to deliver richer functionality and great user experience without having to spend valuable time and resources developing backend server infrastructure and plumbing.

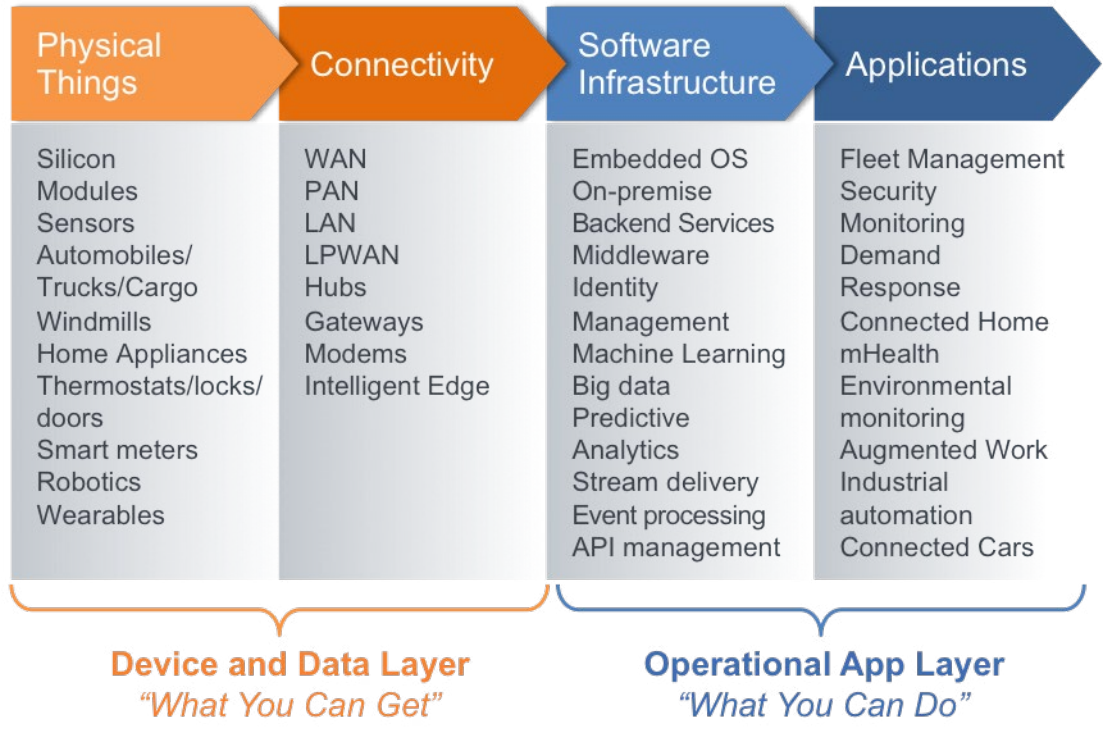
With Internet of Things (IoT) becoming mainstream, the mobile developer community is gearing up to deliver solutions involving a new breed of devices, services, and data source endpoints. Given the similarities between mobile applications and IoT solutions, they can leverage many of their existing skills. Mature MBaaS platforms, such as AnyPresence, have already augmented their platform to support IoT use cases and scenarios, making the transition from mobile application development to IoT development seamless.

With Internet of Things (IoT) becoming mainstream, the mobile developer community is gearing up to deliver solutions involving a new breed of devices, services, and data source endpoints.

Similar to mobile applications, IoT solutions depend on a set of building block services that are essential to address enterprise scenarios. This section covers the key building block components of an IoT app platform.

Borrowing from our friends at 451 Research, let us first take a look at a simplified value chain of IoT solutions. This perspective groups the various components of the overall IoT value chain into four main areas:

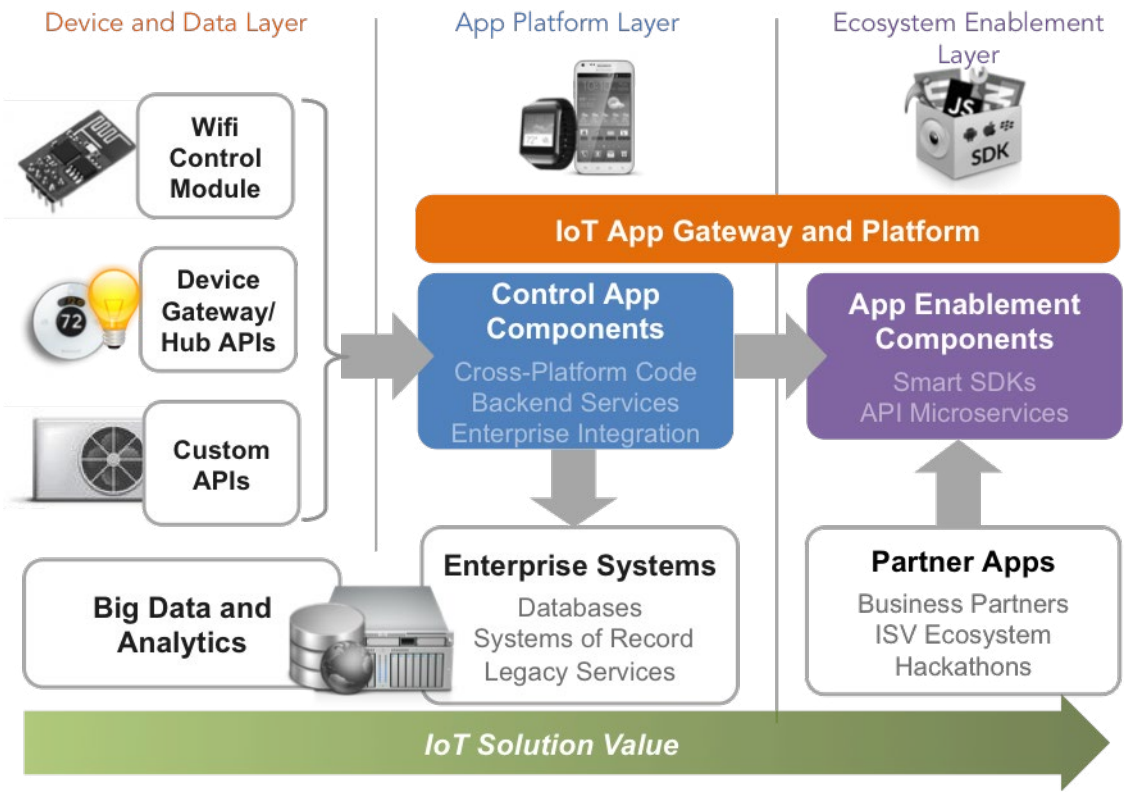
1. **Physical Things** – the physical hardware, sensors, and devices
2. **Connectivity** – the layer that enables connectivity to the hardware
3. **Software Infrastructure** – the layer of supporting software that enables the device to become intelligent
4. **Applications** – the combination and application of the aforementioned layers into a solution that serves a specific need



Source: 451 Research

The first two value chain components can be grouped into what we call the "**Device and Data Layer**", which determines what you can get access to in terms of hardware and raw data or telemetry. The second two value chain components can be grouped into what we call the "**App Platform Layer**", which determines what you can do when you have access to the hardware and the raw source data coming from it.

Building on this perspective, we can then see how these two value chain layers map to an IoT app platform topography, and how a third "**Ecosystem Enablement Layer**" adds another dimension where third party developers can build application solutions leveraging individual components from different vendors.



Devices and Data Layer

Devices represent the very first layer in IoT architecture. This layer includes sensors and actuators responsible for acquiring data or controlling external equipment. Each device node has a microcontroller, a set of sensors, and a communication radio for external connectivity. They are powered by an internal battery or external source. The raw data acquired by each of the device nodes is sent to the next layer that acts as an aggregator.

Device Gateway / Hub

The gateway or the hub acts as an aggregator of sensor data sent by the device nodes. Each device node acquires the raw data which is immediately sent to the gateway. The gateway handles the transformation of data to make it more meaningful by adding additional context. For example, the gateway will convert the raw value submitted by a light sensor into luminance value that will be used for further processing. This layer is connected to the public or private network while abstracting and isolating the device nodes. This isolation adds a security layer that controls direct access to the device nodes. The aggregated and transformed data is sent to the cloud by the device gateway or hub.

App Platform Layer

API Gateway

The app gateway acts as the front door of the IoT platform. Each device or hub sends the data through the API methods exposed by the API gateway. This layer is responsible for translating and routing the calls to an appropriate platform service. The API gateway consolidates disparate APIs exposed by various internal components. It unifies all the APIs by exposing a consistent façade for app developers. The key advantage of using an API gateway is the abstraction it provides between API providers and consumers. It also does the basic authentication of devices. Even when the underlying API changes, the gateway will ensure that the applications always see a consistent interface. This pattern makes it possible to swap the components and functionality without breaking the application.

The key advantage of using an API gateway is the abstraction it provides between API providers and consumers.

Backend Services

Once the App Gateway processes the messages sent by the sensor nodes, they are forwarded to the backend layer that does the heavy lifting around data storage, augmentation, and business logic. The following are example functions that an IoT backend services layer performs:

Authentication

While the API gateway performs basic authentication, the middleware further validates that the device is authorized and authenticated to send and receive messages. This approach adds another layer of security for devices participating in the solution. It also authenticates users interacting with the backend platform and the devices.

Data storage

The storage layer is used for storing both the metadata and data related to sensors and devices. Every device that is registered with the IoT platform will have its metadata stored here. The metadata contains the key attributes such as the unique identifier, device capabilities, authentication and authorization information, and the associated hub information. This layer also stores the raw incoming sensor data sent

by the devices and hubs. It also contains the transformed and processed data after being processed by any custom business logic.

Business logic

Each message sent by the sensors will get processed and orchestrated by the business logic layer. Developers implement code that could do anything from transforming the data to performing lookup operations to executing workflows. The code containing the business logic is typically implemented as microservices where each module is autonomous and self-contained with clearly defined context boundaries. Each function or module is designed to perform a fine-grained operation. By connecting multiple such modules, developers can compose a workflow with multiple stages. With the growing popularity of JavaScript, many of the platform providers host node.js for running the server-side business logic. This enables developers to write and test JavaScript functions and modules on their development machines before deploying them to a production environment.

Events

The business logic layer might trigger events that need to be processed further. Long-running or asynchronous tasks might raise an event when they are complete, and similarly an event might be raised when a data point is above a defined threshold. Developers use the event framework to communicate with the devices and end-users. For example, when the temperature sensor's data crosses a specific value, the developer will wire up the event with the code to turn on an air-conditioner and send a push notification to the user's mobile phone about the operation. Since the event engine is decoupled from the business logic, the platform offers an extensible framework to developers to perform multiple actions for each event.

Extensibility and Integration

The platform further adds an extensibility layer for integrating with third party cloud services, enterprise systems (CRM, ERP), databases, big data processing, advanced monitoring, and analytics. For example, a smart appliance brand may want to integrate with an e-commerce platform to enable ordering of consumables. Or device usage data streams could be forwarded to an Apache Kafka or Apache Hadoop cluster for processing or to Apache Spark for implementing stream analytics to determine if it is time to schedule proactive service appointments before breakdowns occur.

Ecosystem Enablement Layer

The most successful organizations in the IoT space will be those who can enable and foster a strong developer ecosystem. Adoption of a connected appliance, for example, becomes that much easier if it integrates seamlessly and plays well with other components of a smart household. Think of a stove top that can be set to automatically shut off if a house alarm is armed in the away mode.

This type of IoT device-to-device (or M2M) interaction and coordination requires collaboration that can only be realized when each manufacturer opens up access to their devices. The ecosystem enablement layer is what allows partners, customers, developers unlock the power of devices and data through APIs, cross-platform software developer kits (SDKs) and sample app starter kits with pre-built integration to IoT devices. In addition to accelerating the development cycle of the default apps that control the devices, providing these tools to an ecosystem can also help to foster innovation and creativity in ways the product designers never imagined, and are a key component of any holistic IoT app platform solution.

Deployment Models

An enterprise IoT solution leverages on-premises infrastructure and elastic cloud infrastructure for maximum availability. The IoT platform must support running on-premises within the customer's data center, hosted on public cloud, and a hybrid model where certain components of the platform are deployed locally while the business logic and extensibility layer on deployed in the public cloud. The choice of deployment model is critical for deploying industrial IoT solutions.

AnyPresence as an Enterprise IoT App Platform

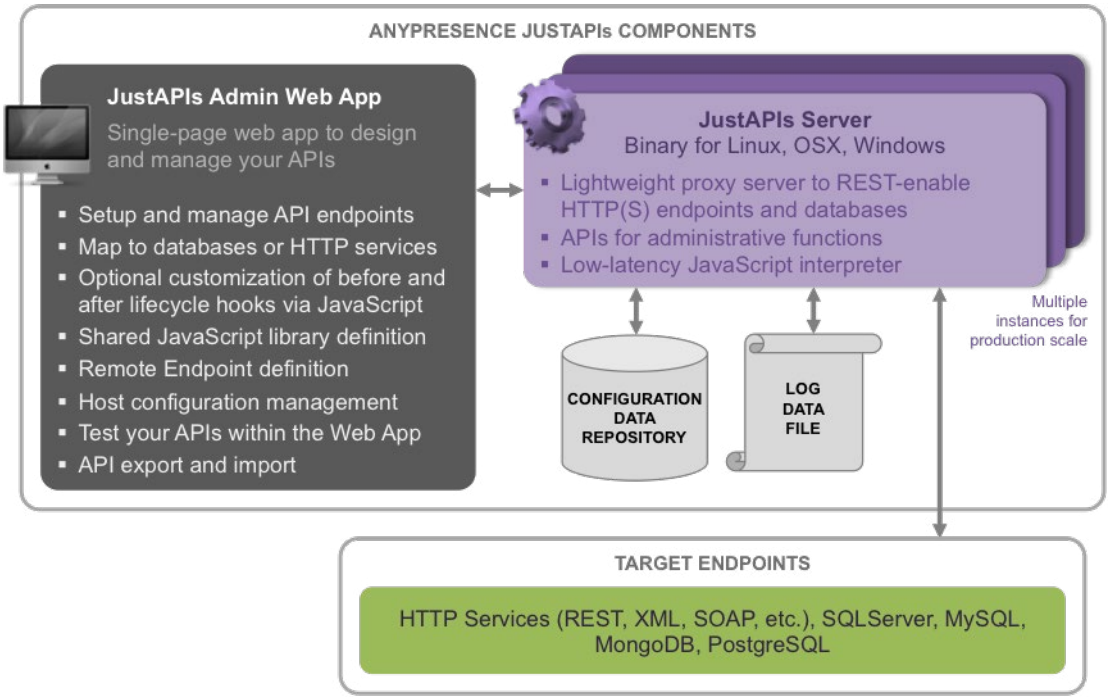
Now that we have established the key components and their roles of an IoT app platform, let us take a look at an example of how an existing platform solution maps to those areas. AnyPresence provides a comprehensive set of accelerants across API creation, app building, and ecosystem innovation enablement. Mobile and IoT Developers can leverage these accelerants without sacrificing flexibility, extensibility, and intellectual property (IP) ownership. The combination of AnyPresence JustAPIs, MBaaS, and App Launchpad offers a comprehensive platform covering a broad spectrum of IoT scenarios and vertical industries.

API Services

JustAPIs is a solution for building and deploying modern REST APIs. As a core component of the AnyPresence Platform, it empowers developers in rapidly defining and implementing custom APIs. It is based on a single, self-contained binary executable that can be deployed on virtually any operating system and hardware, including ARM-based micro-computers.

Custom API endpoints can be defined as simple standalone calls that return a response based on JavaScript logic, or consist of a complex workflow that connects multiple remote web services or databases, and combines the individual results into a single “composite” response. Once developers have defined and tested API endpoints, a production-level environment of clustered JustAPIs instances can be deployed in the cloud or on-premise, with a load balancer or API management solution if desired, to publish your APIs to the world. This capability for rapid API building and flexible deployment model makes it an ideal choice for implementing IoT APIs and playing the role of an app gateway.

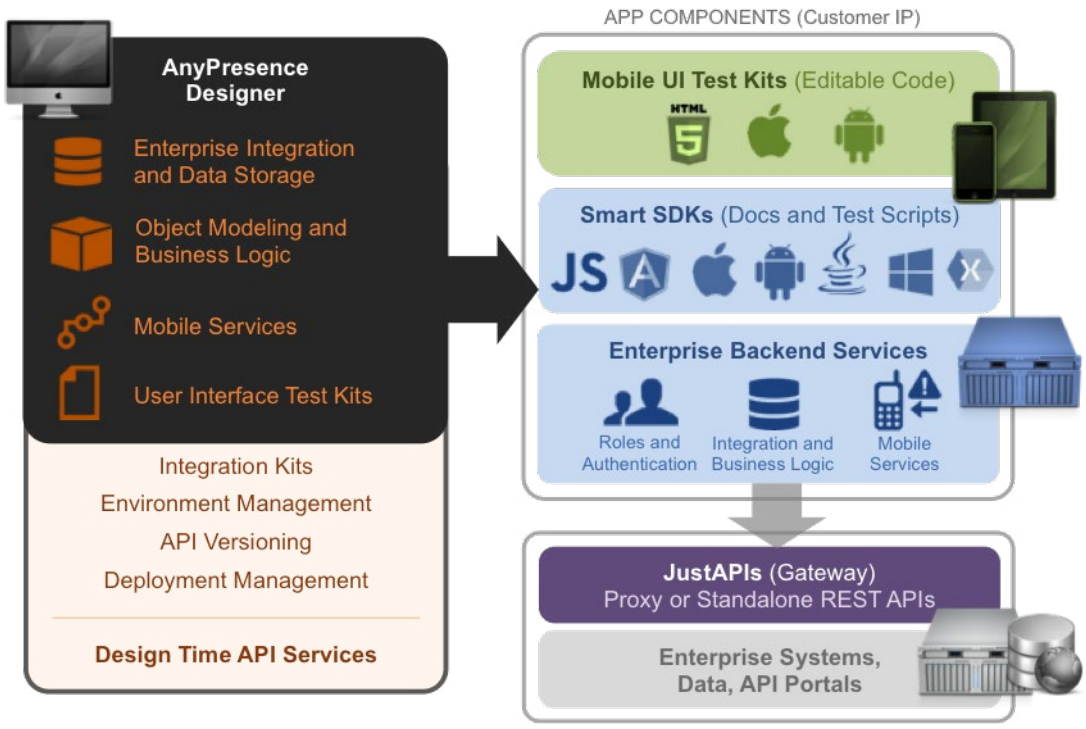
JustAPIs can also be used to simulate IoT device APIs before the hardware exists or is available to developers. API endpoints that simulate the device functions and return sample data or response codes can easily be created for developers to test their mobile or web apps. Once the hardware is available, the JustAPI endpoints can be routed to the hub APIs with minimal changes to the app code.



Backend Services

The core AnyPresence platform provides a robust enterprise-grade backend services layer that can be used to power a broad range of IoT applications. The AnyPresence Designer enables developers to easily define the data source connectivity, object model, and server-side business logic required for an IoT solution. From this definition, AnyPresence then generates the run-time components of the application, including the backend server, client-side SDKs, and even a user interface test kit. These run-time components power the data persistence, integration to source systems, role-based access, authentication, push notifications, and other critical IoT application capabilities.

Unlike other platforms that have many dependencies or customization constraints due to proprietary technology, AnyPresence is easy to use, yet flexible enough to enable source code-level customization and support any deployment model. Most importantly, for manufacturers who prefer to own the intellectual property of companion IoT apps, AnyPresence provides source-code control and ownership for the backend server and client-side code that is generated from the Designer.



Developers can implement custom, server-side business logic using familiar programming languages, such as JavaScript. They benefit from automated generation of RESTful Web Services APIs and corresponding test scripts and documentation. The platform supports multiple versions of server APIs ensuring that the client compatibility is maintained even when upgrading the backend server layer. The source code can be centrally managed in a Github repository.

DevOps teams can use the dedicated administrative web application to view and edit object data, import data, manage users, schedule jobs, view API analytics, manage push notification subscriptions, and perform other run-time tasks.

SDKs and UI Test Kits

In addition to an enterprise backend server, AnyPresence generates a set of corresponding "Smart SDKs" across a broad range of client platforms, including:

- iOS Objective-C
- Android Java
- Windows Phone C#
- Xamarin
- JavaScript backbone.js
- JavaScript Angular.js
- Java

This capability makes it possible to target a broad range of devices and scenarios. Developers can model custom classes with CRUD methods, authentication, deauthentication calls and other helper methods. Test scripts are dynamically generated for all CRUD methods to reduce the time involved in debugging the solution. The SDKs also support offline storage using the native capabilities of each client platform.

In addition to the SDK, AnyPresence can also be used to generate fully-functional, cross-platform test user interface kits, that can be used to exercise the SDK and backend methods without having to build an app from scratch. These apps can be used to test end-to-end connectivity between the client app, the backend server, and the IoT devices without having to write code, enabling the rapid IoT app development and prototyping.

Ecosystem Enablement

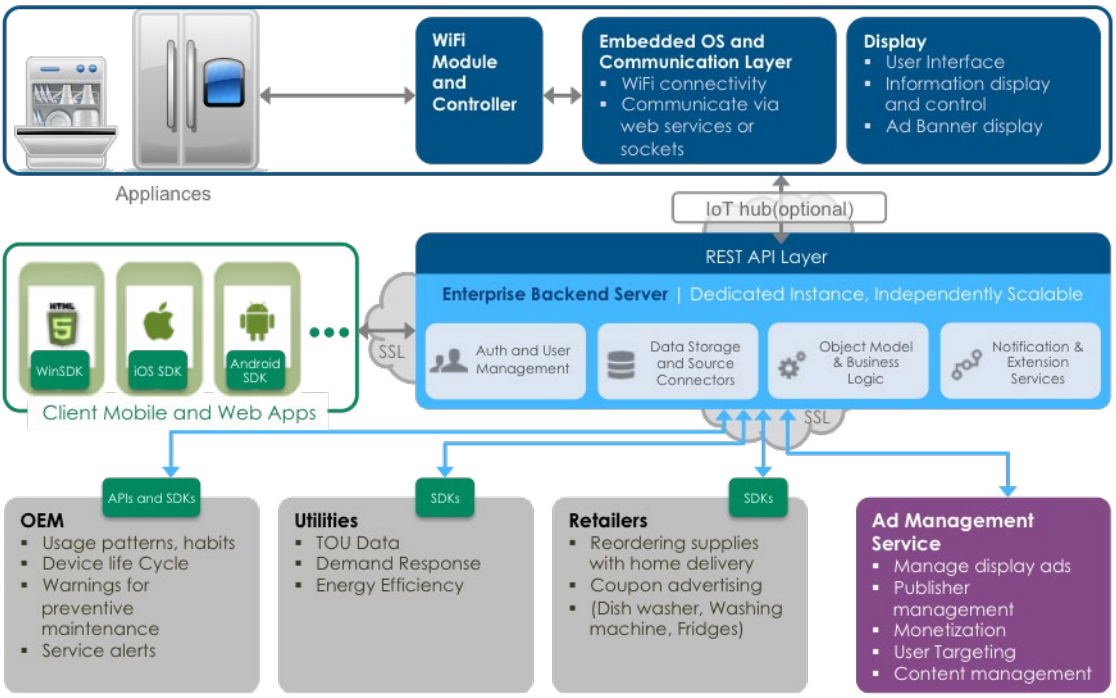
Another unique aspect of the AnyPresence architecture is the ability to for IoT manufacturers to empower an ecosystem of external developers with cross-platform SDKs, user interface kits, test scripts, and even a dedicated backend server running in a sandbox environment with pre-built integration to IoT devices. These components enable third-party developers to quickly build broader IoT solutions that integrate device capabilities from multiple vendors.

AnyPresence App Launchpad can be used to enable a developer ecosystem in the following manner:

1. Define app templates with pre-built integration to the IoT device. Determine which template features should be selectable or configurable by the users.
2. Incorporate into an existing or new developer portal the ability for users to select a template, specify options, and generate cross-platform code
3. Publish the solution so developers can generate and download source code for the backend server, SDKs, and UI kits to develop a fully functional IoT apps with customizable code and deployment options.

Example Scenario: Smart Appliance Architecture

The following figure illustrates how various AnyPresence components can be used to enable a smart appliance application platform:



Web-enabled appliances interact with the AnyPresence backend services layer either directly or through the JustAPIs gateway. Mobile applications that control and manage the devices use the client-side SDKs, which enable faster, more stable app development. The AnyPresence backend server can integrate with third party services, such as utilities to add energy management or demand response functionality, for example. Finally, AnyPresence App Launchpad can be used to provide SDKs to ecosystem partners who can contribute additional services or build their own companion apps.

Summary

The IoT space is evolving rapidly and there are no deeply entrenched, time-tested standards or guidelines for businesses to follow yet. It is becoming increasingly important for organizations on both the supply and demand side of the IoT market to understand how to effectively build apps and enable innovation ecosystems that leverage this connected device landscape. In many cases, a new set of skills is at play and organizations must recognize that they might not have all the capabilities, nor is it a wise use of resources, to build a complete IoT solution on their own from scratch. Leveraging backend services and API gateway platforms can help to reduce the time, cost, and risk involved in bringing sophisticated IoT solutions to market.

A combination of an API builder, enterprise backend services, cross-platform SDKs, and test UI kits deliver everything that a developer needs to build comprehensive IoT application solutions. Companies should look for platforms that offer the most flexibility, enabling them to employ best-of-breed toolkits and frameworks that help accelerate the delivery of IoT applications. They should consider hybrid deployment strategies to seamlessly connect and interoperate with existing business applications, while taking advantage of cloud infrastructure where possible.

Finally, as IoT segments mature, it will become evident that the most widely adopted and successful products are those that support a strong ecosystem. To that end, ensure that your IoT app platform includes the ability to easily publish SDKs and template app starter kits to your developer ecosystem, enabling new and innovative solutions to be built on your IoT product stack and ensure continued success.

Additional Resources

- What to Consider When Selecting a Platform for IoT App Development
- Adobe Think Tank: The Internet of Things
- Top 10 Characteristics of Enterprise Backend Services (MBaaS)
- How to enable an App Ecosystem for the Internet of Things
- Why Use a Platform?
- JustAPIs Developer Edition

About the Author

Janakiram MSV is the Founder and Principal Analyst at Janakiram & Associates. He was Founder and CTO of Get Cloud Ready Consulting, a niche Cloud Migration and Cloud Operations firm that got acquired by Aditi Technologies. He is a Google Developer Expert for Cloud and Microsoft Regional Director in India.

Janakiram is a regular contributor to the Forbes Technology section. Prior to that, he was a senior analyst with Gigaom Research analyst network where he analyzed the cloud services landscape. He is also a contributor at The New Stack and TechRepublic. He is a guest faculty at the International Institute of Information Technology, Hyderabad (IIIT-H) where he teaches Big Data, DevOps and Cloud Computing to the students enrolled for the Masters course.

Janakiram has worked at world-class product companies including Microsoft Corporation, Amazon Web Services and Alcatel-Lucent. His last role was with Amazon Web Services as the Technology Evangelist where he joined them as the first employee in India.