

# Developing Enterprise M2M Apps in Days or Weeks — Not Months or Years

Developers are building M2M applications to help companies reduce service and fuel costs, generate new sources of revenue, and increase customer satisfaction. The market is so hot that market analysts are predicting the number of connected devices powering these applications to reach more than two billion in just a few years. Market segments and applications that are driving the market need for M2M solutions include:

- **Vehicle Telematics and Fleet Management** - Applications in this arena enable users to monitor and track the location, movements, status, and behavior of a vehicle or vehicle fleet.
- **Usage-Based Insurance** - With the simple installation of a hardware unit in passenger vehicles, M2M applications now enable insurance carriers to review driver activity trends and reward drivers based on good driving behavior.
- **mHealth Solutions** - Health is an emerging market where both wired and wireless medical devices are remotely monitored and the data they collect is shared with doctors and patients through portals and dashboards. Doing so helps ensure device uptime and accuracy, increases billable activity and helps improve patient care.
- **Home Energy Monitoring** - Simple applications driven by smart energy sensors enable and disable power outlets, while advanced applications on Smartphones are enabling a broad range of connected devices to provide homeowners with remote control and cost-saving suggestions.
- **Supply Chain Optimization** - Solutions that combine product status at key inspection points are supplying logistics information to distribution managers for interactive, real-time supply chain control.

Clearly, this is an exciting time and as a developer, you are likely looking for ways to get started. This white paper intends to do just that. When you're done learning how you can save time in building M2M applications with the Axeda Platform, check out the M2M App Garage at <http://developer.axeda.com/M2Mappgarage> for videos, blog entries and other helpful resources.

## The Long Way — Do-It-Yourself

You may be thinking to yourself:

*How am I going to build my M2M application? I need to have scalability to support a growing number of devices. And I also need to manage different types of devices and protocols. Should I develop my own protocol? What are the development costs for JEE application engineers? Should I host it on Amazon EC2? Should I do*

it myself? Should I open up a data center? Do I need to learn about Hyper Visor? Should I get a SAN? What about caching? Should I get memory? Should I get solid state drives? What about Rack Space? Are they cheaper than Amazon? How does that pricing model work for Amazon, anyway? What about my database and database schemas? Should I go with a relational database like Oracle? Or am I going to go with a NoSQL database like Hadoop or Cassandra? How can I deploy new complex back-end code? By the way, what data am I storing? How do I communicate with these devices? Are they coming in over a VPN? What about TCP, UDP? What about connections and firewalls? What about proxies, HTTP or HTTPS? What about security? Should I encrypt the traffic? Will it use SSL or AES?

STOP.

These are all valid questions to ask as you consider which technologies you need when architecting an M2M solution. But addressing all of these considerations will take you months - or even years - to sort out and cost you time and money.

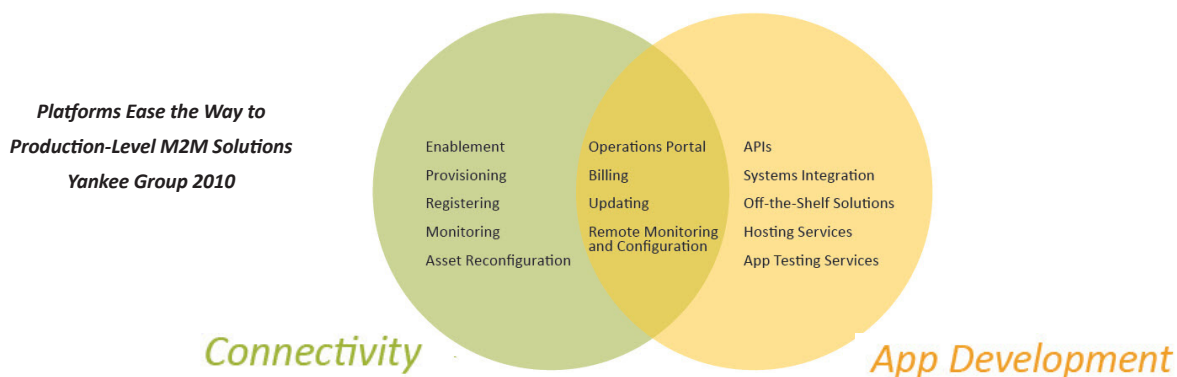
## The Shortcut — M2M App Development Platform

What if you could focus on the value of what your solution provides rather than recreating core infrastructure? Then, you could focus on how your users will use your M2M applications and how they will extract value. What if there was an M2M application development platform to handle hosting, security, scalability, and APIs, so that you could quickly build your apps?

Good news: there is. It's called an M2M app development platform.

Most of us would agree that the term "platform" is one of the most overused and misunderstood terms in the high-tech industry. Rather than offer our definition and interpretation of an M2M application development platform, we'll get an industry perspective from Yankee Group. John Keough's *Focus Report, Platforms: Setting the Stage for M2M Solutions* does a commendable job of detailing the technologies that comprise an M2M application development platform:

While connectivity platforms can drive sets of rules down to individual assets and to some extent, integrate with middleware, they do little to facilitate the development of user-facing applications. That's where application development platforms come in. They provide the back-end support that enables developers to operate assets and build applications. Essentially, they transform raw data streams into interfaces that applications can use to both glean specific device data and allow the end-user to communicate back to the device.



The first set of tools that application development platforms provide are aimed at the operation of the assets themselves. This usually manifests itself in an operations portal that provides visibility into every device in an M2M deployment and allows them to be remotely configured. To further facilitate asset management, most application development platforms also host M2M solutions themselves in secure data centers, or in

some cases, behind a client’s firewall. This takes server upkeep and information security off the list of things solutions providers have to worry about, a feature of special interest to smaller solution providers that may not have enough dedicated IT resources to actively manage an M2M solution’s data flow.

The second main functionality that application development platforms bring to the table is they provide an easy way for M2M solution providers to build their applications. With a wide variance in M2M applications, platforms would be hard-pressed and ill-advised to attempt to provide unique out-of-the-box solutions for all comers in all verticals. Instead, they provide a set of tools to make third-party application development much easier. They provide application programming interfaces (APIs) that allow developers to use common programming languages like Java and Groovy to build solutions. They also offer testing services that let developers test solutions in a simulated M2M environment, allowing them to detect problems prior to the actual device rollout. Through the use of these tools, solutions providers and system integrators can access existing enterprise applications that utilize Web services.

### So, What’s Next?

Now that you have a general understanding of an M2M application development platform, let’s examine some common considerations that apply to most, if not all, M2M solutions:

- **Get Connected** – You need to bring your product or equipment to the network so that it can provide information to the solution, and react to commands and configuration changes orchestrated by the application logic.
- **Manage and Orchestrate Your Data** - Script your business logic in the cloud to tie together remote information with information from other business systems or cloud-based services, react to real-time conditions, and facilitate batch operations to synchronize, analyze, and integrate.
- **Present Your Data with Rich Internet Applications** - Build your user experiences, enabling people to interact with your connected product, manage workflows around business processes, or facilitate data analysis.

Let’s take a look at the services available in the Axeda Platform that help with each of these solution considerations.



## Get Connected

“Getting Connected” is a broad term with different meanings depending upon the environment of your product and the economics of your solution. The Axeda Platform makes no assumption about connectivity, but instead provides features and functionality to help you connect.

For wireless applications, especially those that may use cellular or satellite communications, the speed and cost of communication will be an important factor. For this reason, Axeda has created the Axeda Wireless Protocol (AWP), a cross-platform, efficient binary protocol that you can use to communicate bi-directionally with the platform. While the compact binary format is critical for wireless communications, AWP is also a good fit for connecting devices that have a wired Internet connection. The protocol is expressive, robust, secure, and most importantly, able to be implemented on a wide range of hardware and programming environments.

When you are ready to start coding your embedded agent for your product, the AWP Toolkit can help abstract the communication programming away completely, so that you can focus on the unique aspects of your connected product. The AWP Toolkit is an ANSI C library that can be compiled into your own software and executed on a wide array of computing hardware and platforms. When multiple channels can be used (such as satellite, cellular, and WiFi), the AWP Toolkit supports Least-Cost Routing, where information is sent over a channel based on the information’s value and the channel’s cost. This lets you control the application’s wireless costs, while remaining flexible to business demands.

When you are faced with connecting legacy products that may be communicating with a proprietary messaging protocol, the Axeda Platform can be extended with CODECs to “learn” your protocol by translating your device’s communication format into a form that the platform can understand. This is a great option for retrofitting existing, deployed products to get connectivity and value today, while designing your next-generation of product with AWP support built-in.

## Manage and Orchestrate Your Data

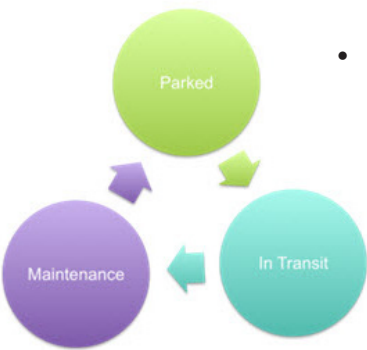
The Axeda data model defines the information and its behavior in the Axeda Platform. It’s important to understand Axeda’s data model when planning to build your applications. However, to keep this white paper within a reasonable length, we would recommend reviewing our technical article on the Axeda Data Model at the Axeda Developer Connection: <http://developer.axeda.com/learn/by-type/technical-article/axeda-data-model>.

## Rules

Rules form the heart of any M2M application.

- **Expression Rules** run in the cloud, and are configured and associated with your assets through the Axeda Admin UI or SDK. These rules have an “If-Then-Else” structure that’s easy to create and understand. They’re like a formula in a spreadsheet. For example, say your asset has a dataitem reading for temperature: if temperature is > 80 then CreateAlarm (“Too Hot!”). This rule compares the temperature to 80 every time a reading is received. When this happens, the rule creates an alarm with name “High Temp” and severity 100.

The screenshot shows a configuration window for a rule named 'TempConvnt'. The 'Type' is set to 'Data'. The 'Description' field is empty. There are two checkboxes for 'Enabled': the first is checked, and the second is unchecked with the label 'Execute action each time rule evaluates to true'. The 'If' field contains 'TempC'. The 'Then' field contains 'SetDataItem("TempF", TempC\*9/5 + 32)'. The 'Else' field is empty. A 'Details' section at the bottom provides instructions on how to use the 'SetDataItem' function, including its syntax: 'SetDataItem(String dataItemName, Object dataItemValue)'. At the bottom right, there are buttons for 'Validate', 'Save and New', 'Save', and 'Cancel'.



- **State Machines** help organize Expression Rules into manageable groups that apply to assets when the assets are in a certain state. For example, if your asset were a refrigerated truck, and you were interested in receiving an alert when the temperature within the cargo area rose above a preset threshold, you would not want this rule to be applied when your truck asset is empty and parked in the distribution center lot. In this case, you might organize your rules into a state machine called “TruckStatus”. The TruckStatus state machine would then consist of a set of states that you define, and the rules that execute when the truck is in a particular state.

## Scripting

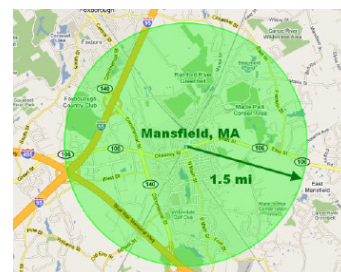
Using Axeda Custom Objects, you can harness the power of the Axeda SDK to gain access to the complete set of platform data and functionality, all within a script that you customize. Custom Object scripts can be invoked in an ExpressionRule to provide customized and flexible business logic for your application. Custom Object scripts are written in a powerful scripting language called Groovy, which is 100% Java syntax compatible. Groovy also offers modern, concise syntax options that make your code simple and easy to understand.

## Location-Based Services

Knowing where something is, and where it has been, opens a world of possible application features. The Axeda Platform can keep track of an asset’s current and historical location, which allows your applications to plot the current position of your assets on a map, or show a breadcrumb trail of where a particular asset has been.

Geofences are virtual perimeters around geographic locations. You can define a geofence around a physical location by describing a center point and radius, or by “drawing” a polygon around an arbitrary shape. For instance, you may have a geofence around the Boston metro area that is defined as the center of town with a 10-mile radius. You may then compare an asset’s location to this geofence in a rule and trigger events to occur.

- IF: InNamedGeofence(“BostonMetro”, location) THEN: Notify...
- IF: InGeofence(“34.5645, -71.3434”, location)...
- IF: DistanceToGeofence(“34.5645, -71.3434”, 10, location)
- IF: DistanceToBoundingGeometry(GetMobileLocation(“10min”) > 100



## Integration Queue

In today’s software landscape, almost no complete solution is an island unto itself. Business systems need to interoperate by sharing data and events, so that specialized systems can do their job without tight coupling. Messaging is a robust and capable pattern for bridging the gap between systems, especially those that are hosted in the cloud. The Axeda Platform provides a message queue that can be subscribed to by external systems to trigger processes and workflows in those systems, based on events that are being tracked within the platform.

## Web Services

Web Services are at the heart of a cloud-based API stack. The Axeda platform exposes Web Service operations for all platform data and configuration meta data. As a result, you can determine the status of an asset, query

---

historical data for assets, search for assets based on their current location, or even configure expression rules and other configuration settings all through a modern Web Service API, using standard SOAP and REST communication protocols.

## Scripto

Web Service APIs simplify system integration in a loosely coupled, secure way, and we have a commitment to offering a comprehensive collection of standard APIs into the Axeda Platform. But we can't have an API that suits every need exactly. You may want data in a particular format, such as CSV, JSON, or XML. Or some logic should be applied, and it's inefficient to query lots of data to look subset you're interested in. Wouldn't you rather make the service on the other side do exactly what you want, and give it to you in exactly the format you need? That is Scripto – the bridge between the power and efficiency of the Axeda Custom Object scripting engine, and a Web Service client. Using Scripto, you can code a script in the Groovy language, using the Axeda SDK and mash up results from other systems, all within the platform. You can also expose your script to an external consumer via a single, REST-based Web Service operation. This approach allows you to create your own set of Web Services that do exactly what you want.

## Present Your Data with Rich Internet Applications

Rich Internet Applications are a great way to build engaging, information-rich user experiences. By exposing platform data and functions via Web Services and Scripto, you can use your tool of choice for developing the application's front-end. In fact, if you choose a technology that doesn't require a server-side rendering engine, such as HTML, AJAX, Adobe Flash, or Microsoft Silverlight, then you can upload your application UI files to the Axeda Platform and let the platform serve your URL!

## Far-Front-Ends and Other Systems

If a client-only user Rich Internet Application interface is not an option for you, you can still use Web Services to integrate platform information into other server-side presentation technologies, such as Microsoft Sharepoint portal or a Java EE web application. You can also get lightning-fast updates for your users with the Enterprise Subscription Service (ESS). ESS allows your far-front-end to subscribe to an assortment of events that relate to an individual or set of assets. While your users are viewing a drill down on a set of assets, they can receive asynchronous notifications about real-time data changes, without having to constantly poll Web Services.

## Axeda Corporation

Axeda is the leading cloud platform provider for connected products and M2M applications. More than 150 leading companies, including Agilent, Diebold, and EMC, rely on Axeda to connect any product on any network, and manage their connected products through the cloud with complete security and scalability —enabling them to innovate with extensive M2M application development and integration capabilities.

By relying on the Axeda Platform to power their connected products, companies are transforming their business by improving customer satisfaction, reducing costs, and generating new sources of revenue. The M2M solutions behind these connected products range from remote service, fleet management, usage-based insurance, asset tracking, mHealth, and more. Visit the M2M App Garage <http://developer.axeda.com/M2Mappgarage> to learn more about how you can build M2M apps with a variety of helpful resources.



25 Forbes Blvd. Suite 3  
Foxboro, MA 02035 USA  
1.508.337.9200

[www.axeda.com](http://www.axeda.com)  
[developer.axeda.com](http://developer.axeda.com)  
[developer.axeda.com/M2Mappgarage](http://developer.axeda.com/M2Mappgarage)