

# Getting HVAC Connected



How to get your HVAC devices connected to open new market opportunities for your business and truly affect your bottom line.



Cesanta Software  
www.cesanta.com  
@cesantaHQ

**Phone:** +353 1 482 44 52  
**Email:** support@cesanta.com

# Contents

<i>Introduction from Cesanta</i>	<b>3</b>
<i>The HVAC marketplace</i>	<b>4</b>
<i>Use cases</i>	<b>6</b>
<i>Solutions that are right for you</i>	<b>8</b>
<i>Defining your technical spec</i>	<b>9</b>
<i>Homegrown vs vendor</i>	<b>10</b>
<i>Selecting the right vendor</i>	<b>12</b>
<i>Technical Spec - Mongoose</i>	<b>13</b>

# Introduction from Cesanta

*The HVAC industry is adopting the Internet of Things (IoT) fast to develop products with a competitive edge. This ebook will guide you through the initial steps you have to take to get connected.*

---

The HVAC industry is changing. No-one can ignore the innovations in the consumer space and the growing demand from users to have access to their air conditioners, heating and ventilation systems via a web or mobile interface. Cost savings from understanding usage patterns and other data as well as getting the most out of energy resources are undeniable. Early leaders in the industry are establishing themselves.



Cesanta is a leader in the embedded communication space. Founded by ex-Google engineers, we work with companies from the HVAC, home automation, manufacturing and other industries to get their devices connected in a secure, reliable and scalable way for their business.

In this ebook we want to take a closer look at the developments in the market and see how these could be applied to different use cases - your business even!

We will give you an objective overview of the options you have to get your devices web ready as well as some technical hints & tips.

This eBook is for you if you are in R&D, Product Management or Software Engineering and are in the early stages of considering moving your product into the IoT (Internet of Things) space or are trying to guide nontechnical team members in the business

# The HVAC marketplace



## HVAC - An Overview of the Market

“

*“Global demand for HVAC equipment is forecast to rise 5.7 percent annually to \$120 billion in 2018. The Asia/Pacific region will remain the dominant geographic market, while North America grows the fastest.”*

*Freedonia Study 2014*

The HVAC industry is in a growth phase with rises in the market predicted as high as 5.7 percent by 2018. In dollars we are looking at \$120 billion.

And while a part of the growth is focused on developing nations, who have themselves experienced economic growth, a large part is based on the replacement of existing systems with new, more energy efficient models and solutions. And there is a large market for tech savvy products.

Manufacturers are looking for solutions that are greener and offer energy savings to their production lines. Connected devices can help predict usage and issues while pushing updates. This brings savings to the environment and the bottom line.

Millennials, those born between 1981 and 2000, now make up about 27% of the US market alone. This is the first generation considered predominantly digitally native, with a spending power of \$600 billion per year / \$1.4 trillion by 2020\*. For this market segment having smart products that understand their lifestyle is a major selling point.

Let's take a closer look at how the industry is already tackling this demand.

*\*Leaderswest*

## The HVAC Industry and IoT Market Statistics and Trends

### MARKET SHARE

**70%  
Share**

2 sectors will account for most of market share: consumer electronics and **intelligent buildings.**

### GROWTH



In 2014 the US and European markets for smart home thermostats **more than doubled.**



By **2022**, a typical family home could contain more than **500 SMART DEVICES.**

**68  
million**

Homes are estimated to be smart by **2019.**

### REVENUE

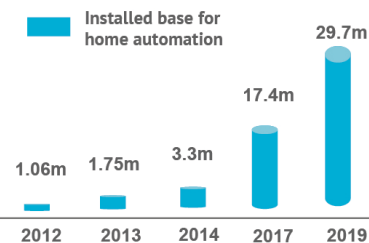


At \$250 dollars per unit, Nest sales estimated to hit **\$300 million annual revenue.**



**European** sales of smart home thermostats increased by **96%** in 2014.

**29.7 million** is the estimated number of the installed base for home automation systems by **2019 in Europe.**



**\$200k**  
potential savings  
PER YEAR

IoT-enabled HVAC systems can offer savings in the **manufacturing field** as well. By understanding expenses and energy usage you can plan for effectively and cost efficiently. GE noted that an improvement of 5% in a small industrial power plant generating 15MW can save over **\$200,000 on average per year.**

### Thanks to our resources:

[Lopez Research and Cisco.com](#)  
[Machine Research](#)  
[Berg Insight](#)

[Business Insider](#)  
[iControl](#)  
[Statista.com](#)



Leader in embedded communication technologies.

# Use Cases

*We've pulled out four common examples of connectivity in the HVAC industry that will impact your bottom line.*

Each use case gives you a concrete example of how an embedded web server can advance your product and even servicing offerings. As we know our own product best, the technical descriptions are based on Cesanta's Mongoose product, you can however still build these functions if you have the capability in-house or choose a different vendor.

## **Use Case 1a - Dashboard Server - Customer Facing:**

From your consumers' perspective this is about being able to access the unit remotely through a web server. This could be by phone or desktop and enables them to decide on the temperature of an air conditioning unit for example while on the way home.

## **Use Case 1b - Dashboard Server - Technician Facing:**

Let's look at the same scenario from the service technicians' side. They could access any unit's event log, history etc before going to service it to be better prepared. You could also add an advanced dashboard to give the technician a more advanced view and ability to access the unit remotely.

Cesanta's Mongoose can serve a dashboard for your unit's control panel, laid out using HTML and JavaScript, with some dynamic and templating capabilities provided by Server Side Includes functionality. For example, a field's value can be set dynamically by invoking C code during page rendering. SSL/TLS can be used to secure the communication and authentication can be handled using built-in Digest authentication mechanism or a custom authentication handler.



### Use Case 2 - RESTful API Server:

Taking use case 1b one step further in a scenario where a service engineer may need to access multiple units at the same time, for example in a hotel, hospital or business premises, this can be achieved as well. This scenario is all about a centralised control panel to handle multiple units remotely.

Here, Mongoose can act as a data source for other external systems by providing data in response to API requests, usually in the form of JSON data.

### Use Case 3 - Real-Time API Server And Server-Side Data Push:

What can be important is receiving actual real-time data when monitoring units to avoid issues during a production process. In a manufacturing plant, you may need to ensure that production machines do not overheat and therefore actual real-time data needs to be accessed and monitored.

Using the built-in WebSockets functionality, Mongoose can act as a source of real-time data for external systems. For example, real-time data for graphs, event logs or alert notifications can be pushed from the server side over a persistent WebSocket connection.



# Solutions that are right for you

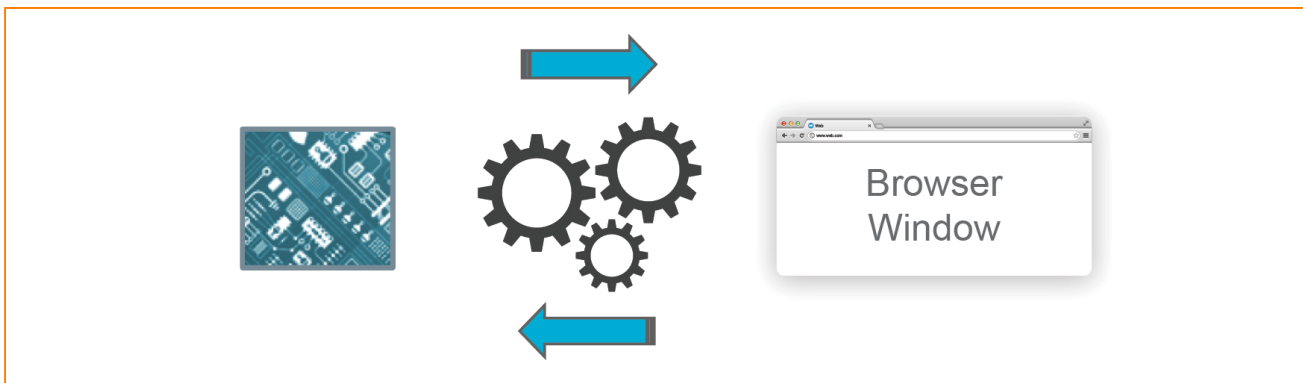
*When it comes to finding a solution to get your device connected, we believe the first piece of software to consider is an embedded, WebSocket-enabled Web Server.*

## Why an embedded Web Server?

The straight forward answer: simplicity. You can build the Web Server into your existing hardware. These can then provide a control panel for configuring your air conditioner, heater or any other HVAC system.

A good Web Server can serve Web GUI on devices, implement RESTful services, RPC (e.g. JSON-RPC), asynchronous Websocket communication and handle telemetry data exchange amongst other functionalities. It should be plug & play for developers: taking existing code and just by adding some additional lines of code be able to integrate it into your existing product.

All the use cases we showed you earlier can be replicated using a Web Server Solution.



## What are the technical specifications that I need to be aware of?

Check out the following page for the questions you need to ask before deciding on the right level of a solution.

## What are my options in the embedded Web Server market?

Essentially you have two options:

1. you can have your team build an embedded Web Server
2. you can go with an existing vendor

We'll dive into the details of each of these options next!



# Defining your technical spec

*Let's take a closer look at the questions you need to ask and aspects you need to consider before selecting a solution.*

---

## **Will the unit be a part of a larger installation that will be centrally controlled?**

If so, be sure that you choose a solution with RESTful API server capabilities and have the internal knowledge base to build / manage this.

## **Does the unit need to be remotely monitored?**

If you need to remotely monitor real-time data, then you need a real-time API server and serviceside data push to make this happen. You also need to think about how the data should be outputted and include this in your requirements for your embedded solution.

However, if you need to access the unit but do not need to see real-time data, then a simpler dashboard server will be the solution. Watch out here for SSL/TLS for security and select the right authentication mechanism.

## **Does the unit have embedded software? If yes, how will it be updated?**

Be sure that the solution you choose allows you to remotely push upgrades and gives you full access to the unit.

## **If the unit is not networked during normal operation, would a service dashboard be useful?**

If you don't want your unit to be online at all times but rather just have it accessed as a technician services it, this is possible and what you'll need here is a strong dashboard server.

# Homegrown vs Vendor

## Building your own solution

Quite often organisations decide to build their own software solutions. We are all aware of bespoke and custom built CRM or CMS systems, homegrown integrations and APIs. Of course, it is also an option to build your own embedded Web Server. There are - as with everything - pros and cons you need to be aware of.



### It's yours

The biggest benefit of taking this route is that the solution you develop is bespoke to you and your product.

If you have a development team in-house with embedded technology skills, this option will be open to you.

With a homegrown solution you can always take it in the direction that you want it to. It will grow as you need it to, no waiting around for upgrades dictated by others.



### Time is money

If you don't have an in-house development team skilled in embedded technology, we recommend staying away from in-house development. The cost of hiring and / or upskilling will eat into your time and heavily into your budget.

You also won't need to pull your team away from other projects to work on this and can rely on the expertise of specialist software companies to develop the software for you. Especially when implementing web technology is not your core business.

In-house solutions can tend to age badly in that they are seldom updated and tend to lack behind in the latest technology. If you are going for this option, you have to fully commit your team to maintaining and updating the software as the technology develops.

## Integrating a solution from an existing vendor

If you don't have an in-house development team or you don't want to move your existing team away from their day job to take on the development of a new piece of software, then going with an existing vendor will suit you better. There are many more pros and cons to consider:



### Experts

When you choose the right vendor, you are dealing with experts in the field who will update the technology regularly and make sure you are not lagging behind.

You will be working with a tried and tested embedding API.

The cost of purchasing a web server is going to hit your budget minimally compared to the cost of time for your team to build and maintain a homegrown solution.

The market is racing ahead - going with an existing vendor will allow you to plug & play, rather than spend a lot of time in development.



### The wrong partner

Working with a vendor should be easy. Firstly, the software should be self-explanatory so that your team can easily integrate it into your device. However, not every project is straightforward and not every vendor offers support.

You want to be sure that the solution you select is lightweight and therefore doesn't impact heavily on the device you integrate it with.

Future proofing a partner is also important. You want to be sure that the partner you choose has a track record of delivering stable solutions.

# Selecting the right vendor

*If you decide to purchase an embedded Web Server Solution, be sure that you are going with the right vendor.*

Choosing the right solution is key to your success. You want to be sure that any vendor you deal with covers the items listed below:

## *Size*

You are integrating with your existing solution and need the most compact and lightweight solution out there to give you room to maneuver.

## *Security*

Ensure you can use SSL/TLS and have an authentication mechanism in place if the units are accessed remotely.

## *Stability and Maturity*

Go with a solution that your peers trust and has been around the block a few times. Check GitHub and other communities to get the low-down.

## *Agnostic*

Choose a solution that is platform agnostic. So whether now or in future you want to make a move and develop on eCos, UNIX/Linux, Windows, MacOS or take things into the app space with iPhone or Android your selected solution will let you do this.

## *Simplicity*

Plug & play. Use a library that you can easily integrate by pasting the code and just adding a few lines of code. Choosing a vendor should make your life easier.

## *Clarity*

Look for clarity when checking out the embedding API. Ideally the source should be in a single file to make embedding easy.

## *Licensing*

Clear pricing that suits your needs. Start with a product that is open source so you can test and when you are ready to commercially apply the solution, change to a commercial license. Ideally you want a vendor that understands that the size of the project matters in pricing: are you using this for one product line or more?

## *Support*

You want a solution that is simple and straightforward so that ideally you won't need support. But especially if this is the first integration project, the option to fall back onto a support structure is comforting and will ensure you can move your project forward fast.

# Technical Spec - Mongoose

Cesanta has been working in the Embedded Web Server space since 2004. Our Mongoose software is trusted and used by the world's largest technology companies.

## Let's have a look at the basic specifications:

- Cross platform: eCos, QNX, UNIX/Linux, Win, MacOS, iPhone, Android and others
- WebSocket, CGI, SSI, SSL, Digest auth, WebDAV, resumed download, URL rewrite
- HTTP proxy, custom error pages, virtual hosts, IP-based ACL, Windows service, HTTP/HTTPS client
- Simple and clean embedding API. The source is a single file to make embedding easy.
- Asynchronous, nonblocking core supporting single- or multi-threaded usage

*"The simplicity and reliability of Mongoose convinced us to choose it as the right web server technology in our product."*

*T. Schmidt, Program Manager, GBS*

## Let's see how you could apply Mongoose:

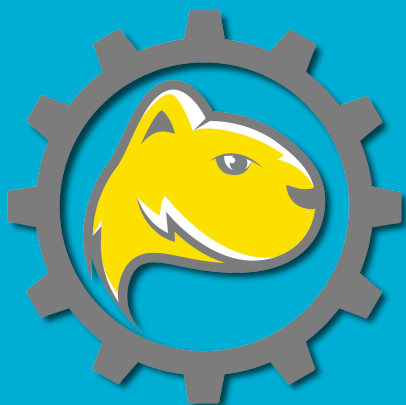
This is an example showing what a dashboard built with Mongoose could deliver for an air conditioning company:

The screenshot shows a web dashboard for 'Acme AC' with the following features highlighted by callouts:

- Secured with SSL/TLS**: Indicated by the 'https://192.168.1.11' in the browser address bar.
- User authenticated via Digest Authentication**: Shown by the 'User: Admin' dropdown in the top right.
- Actions handled by CGI or C functions**: Pointed to the 'Fan Speed: + -' controls for Zone 1 and Zone 2.
- Real-time graph and event data over WebSocket connection**: Pointed to the temperature and fan speed graphs for Zone 1 and Zone 2.
- Static HTML/CSS**: Pointed to the overall layout and styling of the dashboard.
- Simple variable substitution by SSI**: Pointed to the 'Model: AAC1', 'FW Version: 1.1.0', and 'Mode: timer' fields in the right-hand status panel.

The dashboard itself displays two zones (Zone 1 and Zone 2) with their respective settings (15°C and 25°C) and fan speed controls. It includes real-time graphs showing temperature and fan speed over time. Below the graphs is an 'Activity Log' table with entries for 'RPM too low' events, and a status panel on the right showing system details like 'Model: AAC1', 'FW Version: 1.1.0', and 'Mode: timer'. A sidebar on the left contains navigation options like 'Board', 'Tools', 'Settings', and 'Logout'.

[DOWNLOAD THE MONGOOSE OPEN SOURCE CODE](#)



## Mongoose Embedded Web Server - Make any device browsable

- *Trusted and used by the world's largest technology companies & OEMs.*
- *Available since 2004 with over 1 million cumulative downloads.*
- *Extremely lightweight with a core of under 40kb.*
- *Stable, mature and tested.*

Test Mongoose today  
**DOWNLAD**  
THE OPEN SOURCE CODE



**CESANTA**  
Embedded **Communication**

Cesanta Software  
www.cesanta.com  
@cesantaHQ

**Phone:** +353 1 482 44 52  
**Email:** support@cesanta.com

*Thank you to all the great researchers, companies and sites that have made HVAC industry data available online:*

---

[\*Lopez Research and Cisco\*](#)

[\*Machina Research\*](#)

[\*Berg Insight\*](#)

[\*Business Insider\*](#)

[\*iControl\*](#)

[\*Statista\*](#)

[\*Leaderswest\*](#)

[\*Freedonia\*](#)

[\*ReportLinker.com\*](#)