macchina.io
empowering things

# A Guide to Building an IoT Platform

Scalable, reliable, secure

*According to Gartner, 5.8 billion endpoints were in use in 2020, an increase of 1 billion from 2019.*

# The Rise of the Internet of Things

The Internet of Things (IoT) has opened the door to new opportunities and innovations for a huge range of businesses and as a result, the number of connected IoT endpoints around the globe is increasing rapidly. The data being collected has a wide range of uses from monitoring and controlling mechanical devices to optimizing user experience and much more.

This relatively new ability to capture data, often directly from consumer actions, is changing many old business models and fueling a wide variety of completely new business concepts.

# IoT in the Real World

There is a wide range of business models that are leveraging the IoT and this has led to a comparably wide range of IoT platforms, each with their own unique set of features. The following examples identify some of the more common use cases.

## Smart device vendors

These are often agile companies and startups who are delivering data directly to consumers. These companies are equipping their products with real-time sensors or remote control and integrating with consumers' smartphones or other devices. They are often offering consumers access to information about their own behaviour or physical performances or direct control over various electronic devices including vehicles and appliances.

## Transport, shipping and agriculture

A wide range of small and large scale logistical operations are building IoT platforms to help them to monitor their vehicles and machinery to assist with cost optimization and logistics. Among other things, these platforms are used to track locations, predict maintenance of equipment, assess production efficiency, safety monitoring and delivery tracking.

## Industrial plants and factories

Manufacturing facilities and industrial plants are finding more efficient ways to optimize their plant's output and maintenance schedules with a range of IoT sensors and controllers that help to monitor and adjust micro processes. These platforms are also being integrated with digital twins to further improve plant operators' understanding of the inner processes of their facilities.

## Customer care

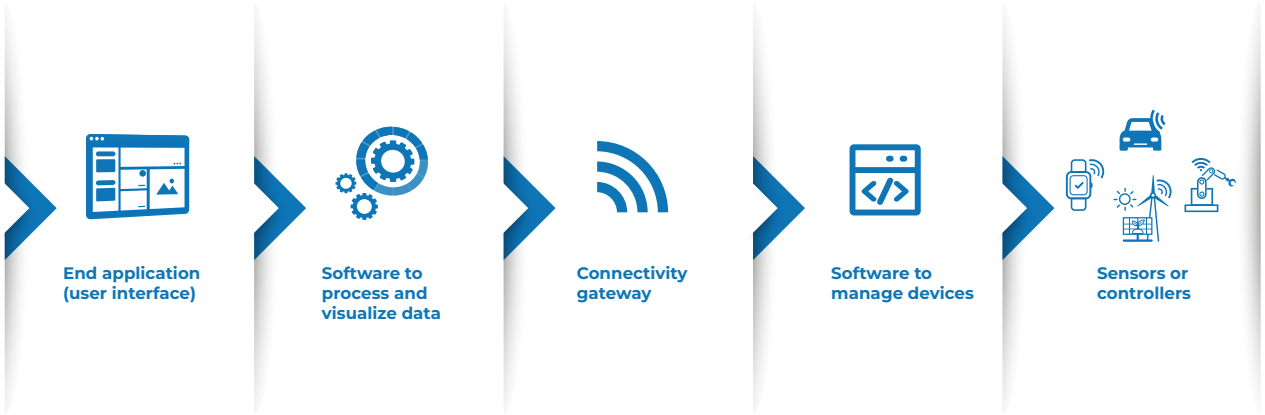
Another type of IoT platform is customer care. These platforms allow companies to interact with people remotely and at the same time collect important behavioural data that further improves their ability to satisfy their customer base. These platforms are being used in retail, hospitality and healthcare.

## Smart cities, telecommunications and energy

At the larger end of the scale, companies and governments are building platforms to help better manage resources in our day to day lives. Most of these are being patched together by combining a range of old (existing) platforms and new platforms. The focus of these platforms is usually utilities and major infrastructure. They monitor and control things such as water, energy consumption, air quality and traffic just to name a few.

# What is an IoT platform?

BASIC PLATFORM COMPONENTS




The IoT platform is the software that allows a business to monitor and sometimes control each device or IoT endpoint. It could be an on-premise suite or cloud hosted service and depending on the business model may have a range of other features and services.

A traditional IoT platform is middleware that provides all of the services and connectivity between the hardware and the applications that need access to the hardware.

The specifics of the IoT platform will vary based on the business model but generally speaking there are some major components that are considered fundamental building blocks.

*In IoT platform needs to securely register, organize, monitor, and remotely manage IoT devices at scale*

## Sensors or controllers

These are what actually collects the data and/or reacts to the commands that are being sent to the device. They are usually built into the hardware (products and devices). An example could be the GPS in your phone or the fuel gauge in your car.

## Connectivity gateway

The IoT device needs to create a secure connection to the IoT platform. This allows data to be sent, and commands to be received. The method of connection might be WiFi, Bluetooth or connecting directly to the internet, but whatever solution you use the device requires software that can open a connection to your IoT platform in a secure and reliable way. Protocols like MQTT and HTTP (REST) are frequently used to connect the gateway to the IoT platform.

## Software to process and visualize data

Once the data is sent to the IoT platform it will need to be processed and interpreted. The specifics of this will vary greatly depending on the product and or service. This may include technologies like statistical analysis and machine learning in order to quickly detect anomalies. Also dashboards visualizing aggregated information in an easily comprehensible way are often part of such a system.

## Software to manage devices

With hundreds, thousands, or even millions of devices in the field, manually managing these devices is simply not feasible. Some IoT platforms therefore include a device management component


that typically provides features like monitoring the health of each device (and sending out a notification if a device starts to fail), updating the device configuration, remotely accessing the devices in a secure way, and updating the software or firmware of the devices. The ability to easily and reliably deploy new software versions to field devices is a critical feature, in order to fix critical bugs, vulnerabilities or to extend the device with new features.

## The end application (user interface)

This is where the human user can interact with the device in some way. I might just be to read data, such as a temperature, or it might be to send commands back to the device. For example to adjust the settings on a thermostat to reduce or increase the temperature in a heating system.

*Security is often cited as one of the biggest challenges in building and maintaining an IoT platform.*

# Achieving secure connectivity

Secure remote access to IoT edge devices is one of the fundamental building blocks of the Internet of Things. End users want to access and manage their devices via web or mobile app, service partners need access to devices installed at remote locations, and product support teams need to be able to log-in to devices installed at customer sites.

Web-based user interfaces are standard in IoT edge devices and connected embedded systems. They are used for configuration, control and monitoring of devices from PCs, smart phones or tablets. Modern web-based user interfaces are powerful, visually attractive and easy to use. Since their only requirement is a HTTP(S) connection between the web browser and the web server running on the device, they are perfectly fitted for remote access scenarios.Secure remote access to IoT edge devices is one of the fundamental building blocks of the Internet of Things. End users want to access and manage their devices via web or mobile app, service partners need access to devices installed at remote locations, and product support teams need to be able to log-in to devices installed at customer sites.

However, for this to work, the web browser on the client PC or mobile device must be able to create a network connection to the IoT device's web server. This is only possible if the IoT device is located in the same network as the device running the web browser, if the networks containing the client and server are linked, or if the IoT device can be directly reached over the internet. Unfortunately, this is rarely the case in practice. IoT edge devices in the field are often connected to private networks behind NAT routers or firewalls. This is especially true for industrial IoT devices, which are typically located behind a NATrouter.

Furthermore, devices connected to a mobile 4G/LTE network in most cases do not have public IP addresses and thus are not directly reachable. This means that while these devices can open connections to servers on the internet, it is not possible to access the device's web server from the outside, unless additional measures are taken.

Port forwarding and Virtual Private Network (VPN) are widely known and established technologies for enabling internet-based remote access to computers and network devices behind NAT routers or firewalls. However, as detailed in the table at the end of this white paper, both technologies have severe drawbacks related to security and complexity, especially when being used with IoT edge devices.

This challenge led Applied Informatics to create a new technology that offers an effective alternative to port forwarding and VPN. It uses a HTTPS-based tunneling protocol based on WebSockets and creates a secure and reliable connection between the device and the platform. This solution is part of the macchina.io Remote Manager solution and is described in more detail at the end of this paper.

# Selecting a technology pathway

One of the first decisions that you and your team will need to make on the way to building your IoT platform is selecting a starting point. There are four options available: Open source, Software as a Service (SaaS, Licensed software or building it all yourself.

There are many factors to evaluate and they will depend on your budget, the size and skill level of your team and your available time frame.

## Open Source

Open source software can kickstart your project and greatly reduce costs and time to market by providing a working software package out of the box, but it carries some risks that need to be carefully considered before investing time, money and energy.  The most important aspect of open source is the community. While it can be extremely helpful in terms of support and software updates, it relies to a large degree on the popularity of the software. Many open source software projects have lost popularity and faded away due to lack of interest. If this happens you can be left with the huge task of maintaining the codebase on your own. Another possibility is that the project forks and splits into two or more different directions. This may force you to make some very tough decisions about the direction of your own platform.

## Software as a Service

Software as a Service offers a ready to go solution that will greatly reduce development time and may provide a viable low cost option in the shorter term. Software updates are included in your fees and are usually delivered quite seamlessly. Support is also usually very good. The main limitation is usually the flexibility of the platform. Most of the code is not accessible and you may find yourself hitting dead ends as you expand the functionality of your platform.

The low investment in the setup stage of a SaaS solution makes it perfect for a high risk or experimental project that might be discontinued. But for a larger, long term endeavour, the lack of flexibility and control makes it a dangerous option.

## Licensed Software

In many ways buying a software package to deliver your IoT platform sits somewhere between the open source and SaaS. An important difference between commercial software and open source is usually that the commercial software is likely to have a narrower focus which means fewer unwanted features and less code bloat.

One big advantage of buying a commercial product is the predictability. Usually the sales process will allow you to gain a full understanding of the software including all the features and all the costs, before you buy. Like open source, it will greatly reduce your initial development costs, but unlike open source you will usually receive support through the integration phase from an onboarding team to make sure that you get exactly what you need, and to help you plan out exactly where the commercial software sits in your technology stack. Updates will also be included and will be well documented and supported.

## Build it yourself

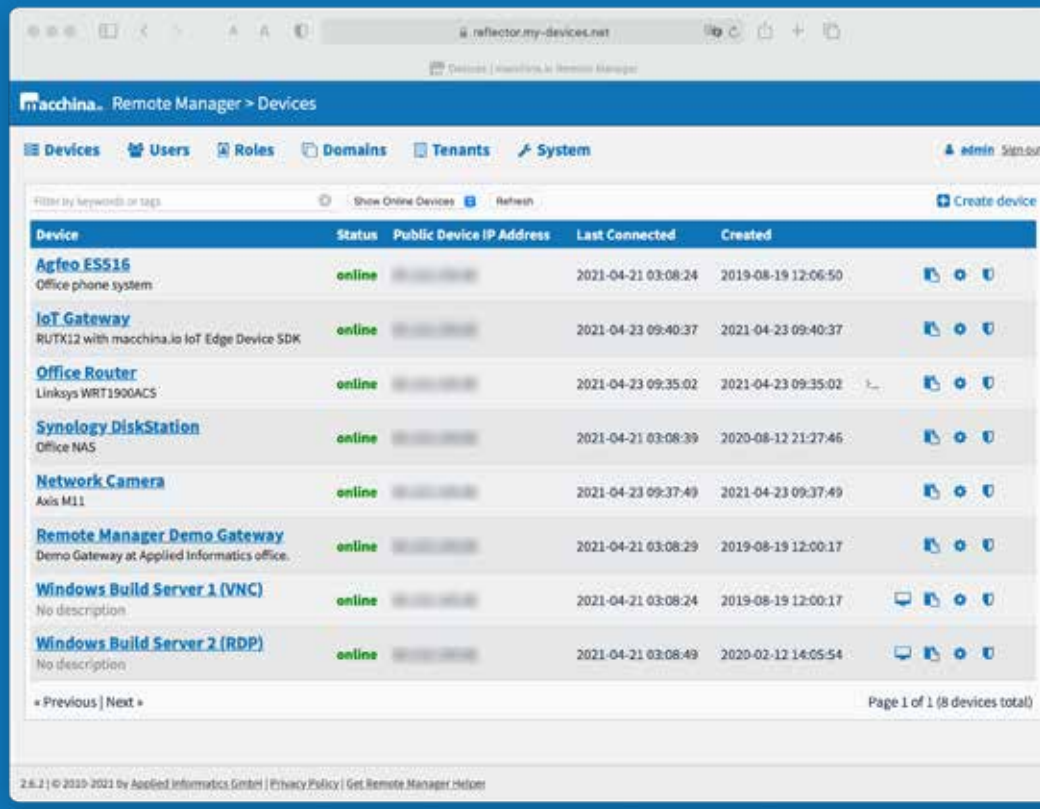
The final option available is to build the entire platform in house. This option is really only viable for very large companies that are developing a system that will scale to a level to warrant a completely proprietary solution. The investment would be significant and would likely require integrating a number of systems to achieve a complete solution.

## SELECTING A TECHNOLOGY PATHWAY

| | Open Source | SaaS | Licensed | DIY |
|---|---|---|---|---|
| Development costs build | High | Low | Medium | Very High |
| Time to market | High | Low | Medium | Very High |
| Maintenance costs ongoing | High | Low | Low | High |
| Support | Upredictable | Good | Very Good | None |
| Flexibility (add or modify new features) | Good | Poor | Good | Good |
| Software updates | Unpredicatble | Easy | Easy | None |
| Security | Unpredictable | Good (tested) | Good (tested) | Untested |
| Scalability | Good | Can be limitted | Good | Good |
| Long term costs (overall) | High | Medium | Medium | High |

# The Macchina.io Remote Manager

macchina.io Remote Manager delivers the core connectivity and data transmission components of the IoT platform while maintaining strict security. It is built for flexibility and can easily be integrated with a wide range of other services.  It enables easy and secure remote access to the web server and other TCP-based services, such as secure shell (SSH) or remote desktop (VNC, RDP), of an IoT device. It works even if the device is located in a private or mobile network behind a NAT router or firewall.




## Web-based Remote Access to IoT Edge Devices with Remote Manager Macchina.io Remote Manager

## Application Scenarios

Remote Manager is built to solve a range of common IoT problems including remote access to:

- IoT gateways, edge computing devices, data loggers and metering and monitoring devices. e.g.renewable energy, environmental monitoring, traffic, transportation and infrastructure
- Remote access to mobile devices for data acquisition, tracking, fleet management, etc.
- Remote support, maintenance and servicing of consumer electronics, home/building automation, HVAC devices, industrial equipment, etc.
- Remote access to IP network cameras and DVRs
- Remote access to security and access control systems.

## How Remote Manager Works

macchina.io Remote Manager is based on standard internet technologies, specifically, HTTPS and Web-Sockets. The IoT device needs to run a program called WebTunnelAgent that opens and maintains a secure, TLS-protected and always-on WebSocket connection to the Remote Manager server running in the cloud. Once the connection between the device and the Remote Manager server has been established, the Remote Manager server uses this connection to send ("tunnel") HTTP requests and other TCP-based network traffic to the device.

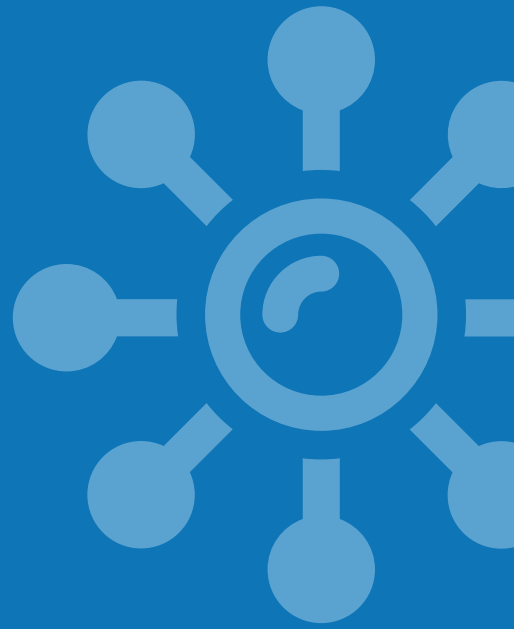
Where do these HTTP requests come from? The Remote Manager server also contains a web server, which accepts requests from clients (web browsers). These requests are then simply forwarded to the device, using the device's tunnel connection. Setting up the initial tunnel connection between the device and the Remote Manager server is almost always possible as long as the device can access the internet. Since the tunnel connection opened by the device uses standard HTTPS and WebSocket protocols, it is very firewall-friendly and even works through an intermediate HTTP proxy server.

## Identifying and Addressing Devices

In a typical usage scenario, more than one device will be connected to a Remote Manager server. In fact, ten thousands of devices could be connected to a single server. Therefore, when the Remote Manager server receives a HTTP request from a client, it needs to find out to which device the request must be forwarded. This is done via the URL sent from the client to the Remote Manager server (e.g., https://dev1.my-devices.net) in the HTTP request. The mechanism relies on a wild-card DNS record in the DNS server which resolves all requests for *.my-devices.net to the Remote Manager server remote.my-devices.net. The Remote Manager server can then use the Host header in the HTTP request together with an internal table to associate the request with a device (and its tunnel connection).

## Running the Remote Manager Server

There are multiple options for running the Remote Manager server. It can be deployed on an internet-facing server in a private datacenter (on-premises), or it can run on a virtual private server (VPS) provided by a cloud service provider such as Amazon (EC2), Azure or DigitalOcean. Running the Remote Manager server can also be outsourced to a dedicated service provider. Multiple Remote Manager servers can run in a load-balancing setup, making it possible to handle 100.000s or even millions of connected IoT devices.



# Security and Privacy Guaranteed

Since the Remote Manager server only transparently forwards HTTP requests and TCP connections, but does not store any data passed through it (except for optional caching of images and style sheets in order to improve performance over low bandwidth network connections), macchina.io Remote Manager does not introduce any additional data security and privacy risks – especially if the Remote Manager server is operated in a private data center.

Of course, both the connection between the device and the Remote Manager server, as well as the connection between the client (web browser) and the Remote Manager server are encrypted and secured with state-of-the-art TLS. A great advantage of this technology is that it is inherently secure. Since the device does not need to have any open ports to the internet, there is no danger of denial-of-service or other kinds of attacks against the device.

Requests to the device can only be sent through the Remote Manager server, and the Remote Manager server requires proper authentication of the user before forwarding requests to the device. Also, devices must authenticate themselves against the Remote Manager server when setting up the tunnel connection. Device authentication is done through a shared secret or certificate.

## CONNECTIVITY TECHNOLOGY PROS AND CONS

| Technology | Advantages | Disadvantages |
|---|---|---|
| Port Forwarding | ▪ simple and widely supported by NAT routers<br>▪ allows access to any TCP or UDP-based network service provided by the device | ▪ NAT router configuration for port forwarding can be complex, espe-cially if multiple devices must be accessible (every device needs a unique public port number)<br>▪ a Dynamic DNS service is needed if the NAT router does not have a static public IP address<br>▪ public IPv4 addresses are becoming scarce<br>▪ the device is directly exposed to the internet – very high risk and danger of denial-of-service and other kinds attacks |
| Virtual Private Network | ▪ the device is directly integrated into a remote network using a secure tunnel through the internet<br>▪ secure, encrypted connection ▸proven, standardized and widely available technology | ▪ VPNs may be blocked by network provider or legally restricted<br>▪ necessary network and VPN server infrastructure is difficult to setup and to maintain, especially if lots of devices must be integrated<br>▪ all clients must have access to VPN in order to access the devices – there-fore not suitable for end-user access<br>▪ additional measures must be taken to isolate devices in the VPN from one another and to prevent users from accessing devices they should not have access to |
| macchina. io Remote Manager | ▪ based on proven and proxy/firewall-friendly WebSocket protocol<br>▪ can be used without changes to the existing network infrastructure<br>▪ supports secure, encrypted (TLS) and authenticated connections<br>▪ secure forwarding of most TCP-based protocols, not just HTTP, in-cluding SSH for remote shell and VNC/RDP for remote desktop access<br>▪ the Remote Manager server can be operated in the cloud<br>▪ high scalability, up to ten thousands of devices per Remote Manager server instance (multiple servers can be clustered to increase capacity up to millions of devices)<br>▪ integrated user management and detailed role-and permission-based access control | ▪ macchina.io Remote Manager agent software must be integrated into device, or a gateway device must be used to integrate legacy devices<br>▪ some TCP-based protocols cannot be forwarded (e.g., FTP)<br>▪ cannot be used with UDP-based protocols |

## User Accounts, Roles and Permissions

The Remote Manager server supports user account management features and role- and permission-based access control, making it easy to specify which users may access and manage which devices.

## Works for Web, SSH and Remote Desktop

macchina.io Remote Manager is not just for accessing web pages. Virtually every TCP-based protocol can also be used over a Remote Manager tunnel connection, including web services based on REST, JSON-RPC or SOAP technologies, or secure shell (SSH) and remote desktop (VNC, RDP) protocols. Remote Manager even includes a web-based VNC client. This makes it a great foundation for automated device management applications and remote support/maintenance portals.

## Easy Integration and Customization

The software necessary for integrating Remote Manager into a device, as well as the Remote Manager server is provided by Applied Informatics. For devices where the necessary modification of the firmware is not possible or feasible, a low-cost gateway device can be used to connect the device to the Remote Manager server. The gateway is located in the same local area network as the device, and forwards requests from the Remote Manager server to the device's web server. It's also possible to install the gateway software on a mobile internet router. The Remote Manager server can be integrated with other applications via its REST API. The default web user interface of the Remote Manager server can be customized to match customer-specific needs and visual style.

The Remote Manager server optionally supports LDAP for user authentication.

# Secure Remote Access Made Easy

macchina.io Remote Manager is a great and secure alternative to technologies like NAT port forwarding and virtual private networks to enable easy and secure remote access to IoT devices via web, shell or remote desktop. The technology can be used without touching the existing network infrastructure and is suitable for use with end users, service partners or internal support teams. The necessary Remote Manager server can be operated in “the cloud”, and devices can be eas-ily integrated, either by updating their firmware or by using a special gateway device or 4G/LTE router.

macchina.io Remote Manager can be used for free with up to five devices. For more information as well as tips for getting started, please visit the website at https://macchina.io.

GET STARTED WITH A FREE ACCOUNT

Applied Informatics Software Engineering GmbH



Image attributions
Icons made by Freepik, Dave Gandy, Kiranshastry, Gregor Cresnar from www.flaticon.com