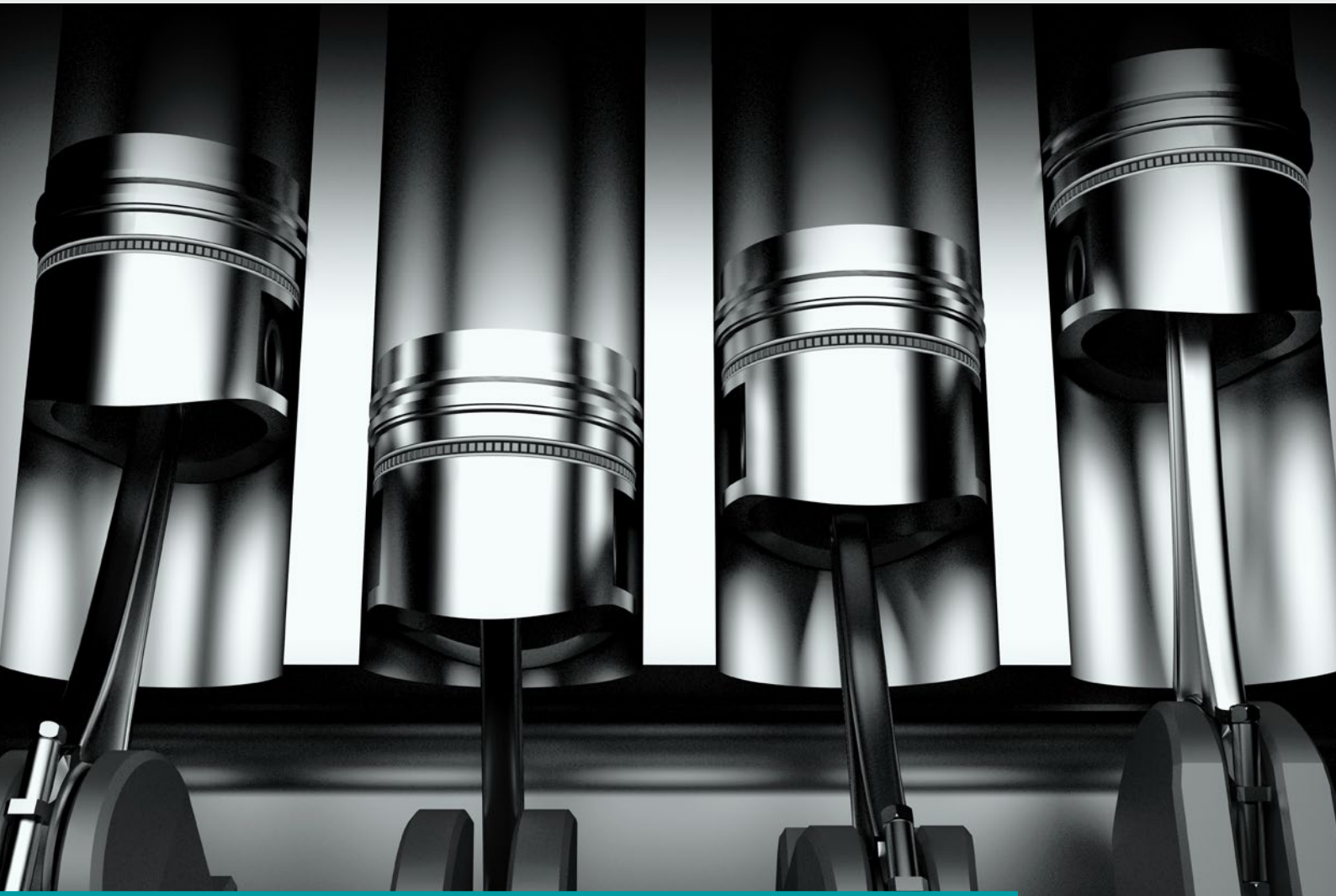
FALL 2018



# KEYnote 36

THE WIBU-MAGAZINE

## Automatic Protection for Your Software

**Highlights**

- Licensing with the Cloud in Mind
- Simpler License Delivery with Push Updates
- License Server in HA Environments

WIBU SYSTEMS

## Content

**LICENSING**
Licensing with the Cloud in Mind 4

**LICENSING**
Simpler License Delivery with Push Updates 6

**LICENSING**
License Server in High Availability Environments 8

**SECURITY**
Acquiring Trust with Mini-Certificates 10

**PROTECTION**
Automatic Protection for your Software 12

**RESEARCH**
Progress on the IUNO Research Project 14

**HIGHLIGHTS**
News in Brief 16

**CASE STUDY**
Success Story | Metrohm 17

**INFORMATION**
Wibu-Systems informs 18

# Dear Clients and Partners!

With the arrival of Fall, I find myself reflecting upon our recent customer successes and marvel at the innovative ways in which our technologies are being used to address critical applications. At the same time, I'm also occupied with a number of business challenges: Recruitment, internationalization, and regulation in cyber-security.

We strive to find the most qualified people to fill vital positions within our organization. Fortunately, here at Wibu-Systems we work with exciting technologies and offer a vibrant workplace to attract the best and brightest. We also rely upon campus partnerships to find the most talented newcomers to the job market. It's an ongoing process and quite rewarding when we make the right connections.

As a manufacturer in the global economy, we rely upon free and fair commerce. Free trade is the catalyst for generating new ideas, inspiring better solutions, and fueling business progress and prosperity. To that end, we are intensifying our sales efforts in Italy and founding WIBU-SYSTEMS K.K. in Japan to expand our international footprint.

The planned EU Cybersecurity Act is the topic of a whitepaper by the German electrical and mechanical engineering associations, ZVEI and VDMA, and the German technical inspection association TUVs that addresses the challenges and possible solutions for cybersecurity. The market is at a critical juncture where it needs globally recognized guidelines to ensure the safety of products and devices from cyber-attacks. This is particularly important in digitized production processes. Legislators are tasked to find a universal approach that encompasses all types of products and their many different applications.

In this KEYnote, you can read about the inner workings of our CodeMeter Protection Suite, use cases for cloud licensing, mini-certificates, user-friendly push updates for licenses, the work of the IUNO project, and the use of CodeMeter by our client Metrohm. There is also an exclusive look at our high-availability license server.

I am always interested in sharing thoughts and ideas with you. I welcome you to stop by and meet me and my colleagues at one of the upcoming trade shows.

Best regards,

Oliver Winzenried

# ALERT

**One idea at the right time can change everything.**

**Subscribe to our blog**


# Licensing with the Cloud in Mind

Predictive maintenance and Condition Monitoring have become hot topics in industry. It means collecting and processing data from as many devices as possible to make statistical predictions and plan maintenance, repairs, and replacements long before worst comes to worst. But what does it entail, and how can predictive maintenance providers make a profit with their services?

## Collecting Data

In order to reliably predict wear and tear and anticipate possible failures, the system needs to collect and process masses of data. This is usually done via the cloud, to which all connected devices send their log files. The developers of predictive maintenance solutions use this data to create predictive algorithms, not unlike meteorologists for weather forecasts: Historical data is used for projections about future events. Neither predictive maintenance systems nor weather forecasters are 100% accurate, but they can tell you with some certainty whether you need to bring a – metaphorical or real – umbrella. With some certainty, it will not snow in Arizona during the summer. It might in Nepal. Location, then, is one of the many factors that make for accurate predictions.

Collecting and processing data requires much labor and resources, but it also adds real value for the user: More than enough reason for selling such services.

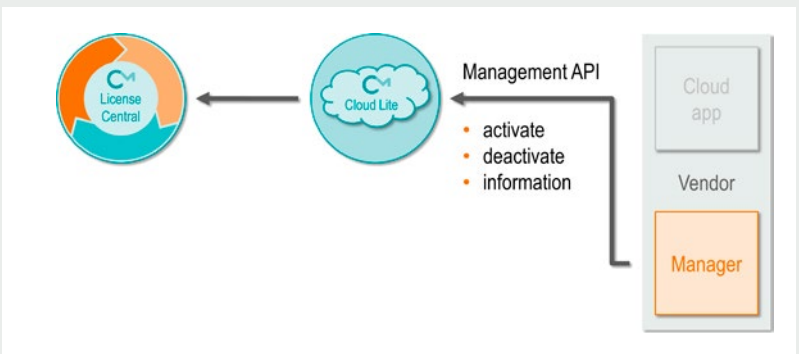
## Licenses on the Ground

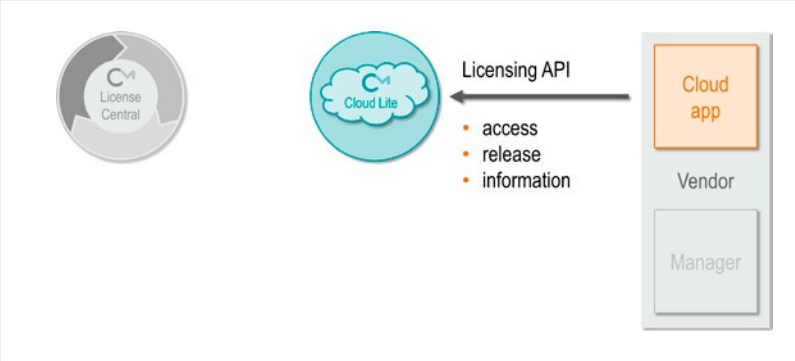
The devices can include industrial controllers in many possible scenarios. Imagine a mining operation: The devices are rarely online, meaning that data could be shared only in sporadic bursts or by physical, offline means. Licensing such hardware is a technical and logistical challenge: The cloud might not be an option a mile underground.

CodeMeter already has the perfect solution for these cases: A CmDongle is connected to the device, or a software CmActLicense is bound to it. The actual licenses can be distributed via the cloud by means of CodeMeter License Central and brought onto the CmDongle or CmActLicense when the hardware comes online again. For even more offline scenarios, you could physically transport a license file to the device.

## Licenses in the Cloud

This type of licensing is the perfect choice for actual devices, but it is less suitable for the mentioned predictive maintenance solution. Its software stays in the cloud, and the license cannot be bound to a specific computer on the ground. You also can not plug a dongle into the cloud. Of course, you could operate physical license servers in a data center and make them available for a cloud solution, but this creates new questions in terms of costs






and availability (which are discussed in more detail in the "TMR Server" article in this issue).

The better option for cloud-based applications is CodeMeter Cloud Lite, a license server that you operate in the cloud for your users. It represents the anchor of trust for the licenses and does so by binding each license to a known user.

## Delivered as Usual

The best thing about CodeMeter Cloud Lite is that it uses the same distribution processes for licenses as are used with CmDongles and CmActLicenses: You create a request with CodeMeter License Central, which returns a ticket as the retrieval code for the license. As the next optional step, the licenses that go with the ticket are then activated on the actual target system.

This activation can take different forms, e.g. with the transfer into a CmDongle, a CmActLicense, or a CmCloudLiteContainer. You, the developer, have the power to decide where licenses can be moved, enabling you to e.g. sell the predictive maintenance engine as an on-premise solution. This might not be necessary in actual business, but it is good to know that you have the ability to do so.

## Binding to a User

By contrast to a CmDongle or an CmActLicense, which have their licenses bound to a device or other hardware, in the case of CodeMeter Cloud Lite the licenses are bound to users. This opens up many other opportunities for simplifying the licensing process for users: Licenses can, for instance, be allocated to the user directly after their creation in CodeMeter License Central. The specific process you employ will depend on your ERP system and established procedures. For instance, you should consider whether you already know the users when an order is created in SAP. If you do, there is nothing to prevent an automated allocation of the license to them.

## What Is Light about CodeMeter Cloud Lite?

CodeMeter Cloud Lite is a simple solution for licensing cloud applications like predictive maintenance. It includes a streamline SOAP and REST-API for a particularly easy integration of license queries in languages like PHP, Java, JavaScript, C#, or other .NET languages. Compared to CodeMeter Runtime, there is no C-interface available as a library, even though a SOAP or REST-API can also be used regularly from C/C++. Relevant examples can be provided on request.

CodeMeter Cloud Lite deliberately comes without the normal key storage and cryptographic features which are indispensable when software is used on premise and in untrustworthy environments. In the case of cloud projects run by yourself, these are unnecessary.

## Available APIs

CodeMeter Cloud Lite comes with APIs for license queries and management APIs for activating and deactivating licenses, which enable you to transfer licenses from CodeMeter License Central to CodeMeter Cloud Lite either automatically or with the simple click of a button on the part of the user. The license is bound to the user as part of that transaction.

You can also deactivate licenses to transfer them back from CodeMeter Cloud Lite to CodeMeter License Central and remove the binding. The deactivated and reactivated license can be moved to another user context.

You can set the management API to start its work automatically when licenses are created in CodeMeter License Central in order to make sure that licenses are immediately bound to their user as soon as they are made.

## Where Did the Server Go?

The CodeMeter Cloud Lite server's home is – unsurprisingly – in the cloud. As it works as an anchor of trust, it should be located at and operated by a trustworthy partner. This can be either you as the developer or Wibu-Systems as your software protection and licensing partner of choice.

As a rule, there is nothing in the technology to prevent you from shipping a CodeMeter Cloud Lite server to one of your larger clients. The question to consider here is the relationship of trust that exists between you and your key accounts. The server could also be operated in a private cloud.

In order to operate a CodeMeter Cloud Lite server, either yourself or at one of your key accounts, you need an unlimited license from Wibu-Systems, which is an annual licensing model that allows you to create an unlimited number of CodeMeter containers during the term of your contract, including licenses for CmDongles, CmActLicenses, or CmCloudLite-Containers.

## Looking Ahead

In early 2019, Wibu-Systems plans to expand the solution with CodeMeter Cloud, bringing the full feature set, including encryption capabilities, to allow servers in the cloud to be used for the secure protection of on-premise solutions. If such on-premise setups are meant to operate offline as well, you can use a combination with CmActLicenses that are distributed via CodeMeter License Central.

Beyond predictive maintenance, CodeMeter Cloud Lite is an ideal choice for many other possible use cases. All applications you are running in the cloud – irrespective of the language they were developed in – can be licensed with CodeMeter Cloud Lite.


# Simpler License Delivery with Push Updates

As of version 6.70, CodeMeter can now create Modified Context Files (*.WibuCmRaM) and combine Update Files (*.WibuCmRaU) for Universal Firm Codes. This sounds like a secondary technical change, but it is indeed one of the major features when it comes to using CodeMeter in the field. This deceptively minor feature allows CodeMeter License Central to cover two new use cases: Delivering licenses without new Context Files (*.WibuCmRaC) and delivering licenses when the Context File is outdated.

## Updating Licenses 101

A few weeks ago, I was invited to a medical device maker who is not yet using CodeMeter. The system currently used by the company works by sending the user a new USB dongle whenever he buys a new feature. If a user pays for two new features, he will receive two new dongles. One feature, one dongle, even if the user buys features in bulk.

This makes it easy for SAP, because the preprogrammed dongles can be delivered as usual. However, the approach does not scale easily, can be quite slow, and simply feels old-fashioned; for this medical device maker, cloud licensing was not an option, because not all of the devices were actually connected to the Internet.

CodeMeter comes with remote programming capabilities on board, based on a file exchange system. The actual licenses are kept on hardware-based CmDongles or in software-based CmActLicenses and can be used without an active Internet connection. To change anything about a license, a Context File needs to come back from the user to you, the ISV. You then create an Update File, which can get to the user through various, even offline channels. This file changes or removes licenses or adds new ones: Exactly as you need it.

## You Only Update Once

Update Files are incremental updates, that is, each builds on the previous update. This means that the chain of updates cannot have a single link missing or added at the wrong time. There are also two other important security features: Every Update File can only be used with the PC or CmDongle that sent the original request. And every Update File can only be used once. This is important for incremental updates, in particular, because a user could otherwise get only one legitimate update and use it multiple times without license to do so. The solution is simple, but restrictive: It all works with one current Context File sent to you by the user. At least, that used to be the case before CodeMeter 6.70.

## Online – Offline No Longer Separate

With CodeMeter License Central, you have a cloud-based solution for creating, delivering, and managing licenses with unparalleled comfort. You give your user a ticket that he can use either in your software or in a dedicated license portal like WebDepot. If your user is online, the Context File is created automatically and sent to CodeMeter License Central, which generates an Update File with the licenses defined for the ticket. The system sends that file back to the software or license portal, where the update is imported and a receipt returned to CodeMeter License Central in the form of a Context File.

Back to our manufacturer who does not yet have CodeMeter on all of his devices. The online capabilities of the CodeMeter License Central cloud solution would make it easy to manage the licenses for all devices connected online. But what about offline devices? In this case, WebDepot allows you to upload the Context File and download the matching Update File; in most cases, the receipt, as the third link in that chain, is optional and can be skipped. For our medical device maker, this was still too unwieldy a process. Let us see how easy we can make it for him.

## Before the First Update

It all begins with your delivery of software or hardware to your user. This normally goes with the purchase of a device or a license, but it could also be done for trial purchases. We can imagine three common scenarios:

1. You deliver your own hardware with an initial configuration of the required licenses, either a demo license, a basic version, or even just an empty container. The container can come in the form of a CmActLicense or a CmDongle.
2. You deliver a CmDongle with a preconfigured license for your user.
3. You send your user a ticket via CodeMeter License Central, which he can use for a CmActLicense or existing CmDongle (with multi-vendor capabilities) on his target device.

In the first two cases, you have the CmContainer with you before its delivery. This is an ideal opportunity to save a matching Context File for each container.

In the last case – the activation on a new device or in an already active CmDongle – the established process would have the user send a fingerprint of this device or dongle to you to get the license in return.

## The Very First Update

After the initial license has been created, you kept a Context File, which you can now use for the first update. In this case, the user does not have to send in the new file, and you deliver the first update by push update instead. Either you or your user simply selects the serial number of the CmContainer that the licenses should go into.

CodeMeter License Central automates this job for you, saving Context Files and Update Files and managing which user goes with which CmContainer.

## The Second and All Other Updates

This is the crux of the matter: We would need a new Context File for the next update, which we don't have. Enter the Modified Context File as the hero of our story: CodeMeter creates a Modified Context File alongside the Update File in the last step. This is the stand-in for the Context File that the user would normally send you, and you can use it now to create push updates for your users without needing anything in return from them. With each update, a new Modified Context File is created for the next update.

From version 3.21, CodeMeter License Central does all of this automatically for you. You select the serial number of the CmContainer under 'Push Update' and download the Update File, or you integrate the same function in your license portal for your users to do this themselves.

## Missed an Update?

But wait! What happens if a user forgets to install an update, or if the update never arrived in the first place? No need to worry: Update Files are made to join up. CodeMeter License Central knows exactly which updates were initiated with a Context File by being notified either by a separate receipt or automatically with the new update. Any update that was not confirmed yet is automatically integrated into the newest Update File for the CmContainer in question. And now for the best part: CodeMeter Runtime will automatically detect which updates have already been installed and skips them. After the new update, the user will be up to speed with whatever happened or did not happen before.

## Everything Automatic

This makes the entire process just as easy for our medical device maker as it used to be with simple dongles. The manufacturer only needs to distribute the Update File that gets everything in order on the client's machine by itself. All the user needs is a dongle or, in the case of CmActLicenses, not even that. Everything is delivered instantly.

One question remains: "What happens if we use the standard path, and a user picks an old Context File by mistake?" Short answer: "It doesn't matter, the update will be handled properly." Longer answer: "CodeMeter License Central has a state counter, the so-called Firm Update Counter, that recognizes that an old file was selected. If it has not happened yet, the process automatically checks off all older updates. The current Modified Context File is then selected, and the update created, with all remaining updates included in the Update File."

## Returning Licenses

Our medical device ISV had one last question in store: "Can you return licenses via push update as well?" I had to ask a counter-question, even though I already knew the answer: "You mean if your users are returning, or you are withdrawing the license?" Of course, both works just as well.

The user will typically return licenses e.g. in order to move to another device. This can then also be done by push update. A new Context File is needed once to confirm the returned license, and the next activation on the next device can again work by push update.

As an ISV, you can also withdraw licenses, which can again be done by push update. If you want to make sure that the recall worked, you can request a receipt in the form of a Context File, but even if you never received the receipt, the user could not skip the push update with the license withdrawal: It will be enforced automatically with the next license update for the CmContainer in question. It is virtually impossible for users to go into hiding and avoid withdrawals, which is helpful for subscription or pay-per-use licenses, in particular.


## Conclusions

With push updates, distributing licenses is easy work, especially for offline scenarios. CodeMeter License Central makes sure that the user will always be up to date, even if he skipped some updates in between. Just like our medical device maker, you – the ISV – can sit back and rely on the automatic features of CodeMeter to do the heavy work for you.

# License Server in High Availability Environments

Guaranteeing optimum availability, while keeping a strict count of the purchased licenses, is a constant cause for friction between software makers and their users. With the Triple Mode Redundancy system, Wibu-Systems has a solution to this conundrum e.g. for industrial users that works without disclosing license calls and without having to rely on trust alone.

## A Fictional Example

Dr. Schwabe, head license buyer at a renowned maker of solar panels, has a habit of calling Mr. Zenz once a year. Every time, it is about the same issue: The quality assurance software works just fine, but then there are sporadic problems with reaching the license server. What is the point of licensing, if it causes problems, he wonders? His company is a company that people can trust.

But Mr. Zenz knows better: Only last month, a support incident revealed that a company used more licenses than it had paid for. The goodwill licenses made available to cover for some recent server issues almost led to a major loss of valid revenue.

## Good Answers

This year, Mr. Zenz finally had an answer that Dr. Schwabe could not ignore: Wibu-Systems is offering a solution that combines the best of both worlds:

- High availability: Server outages or other problems do not have to mean a real service disruption.
- License checks: All purchased licenses are reliably available – no more, no less.

As a Triple Mode Redundancy system TMR Server Setup, the concept works with a combination of a 2-out-of-3 licensing concept and tried-and-tested data center technology. Luckily, Mr. Zenz had already migrated the licensing system to the new Universal Firm Codes, which is a precondition for TMR licenses.

## License Structure

Every license is created in triplicate and given a special ID, the TMR ID, as an additional property to go with the license count. The TMR ID is used as a definite identifier for all three licenses. The firm code, product code, and TMR ID need to match for all three licenses to come together and form one TMR license. Ideally, consecutive numbers are used for each new TMR license.

There is no need to test whether other properties of the product items match, as this would only lead to more complications with later updates. However, it helps if all three licenses going with a TMR license have the same properties.

## Similar CmContainers

The three licenses are placed into three separate CmContainers that have to have the same CmActId, as the TMR Server Setup can only allocate CmContainers with the same CmActId to a virtual CmContainer. This virtual container is the only one seen by the user, with nothing indicating that it is a virtual receptacle of three separate CmContainers. Virtual CmContainers have fully configurable, typically random serial numbers with a new mask byte 131, e.g. 131-59885682.

The same approach naturally also works with three CmDongles, which would then also form a virtual CmContainer with a serial number starting with 131.

## The 2-out-of-3 Rule

For a TMR license to be valid and available, at least two of the three related licenses have to be available. If only one of the three is there, the TMR license will not make an appearance in the virtual CmContainer.

A CmContainer with only one of the constituent licenses is of no use to anyone: The CodeMeter-Server would bar a license with a TMR ID from being used in this case. In effect, such a license could only be used in a full TMR Server Setup.

## System Setup

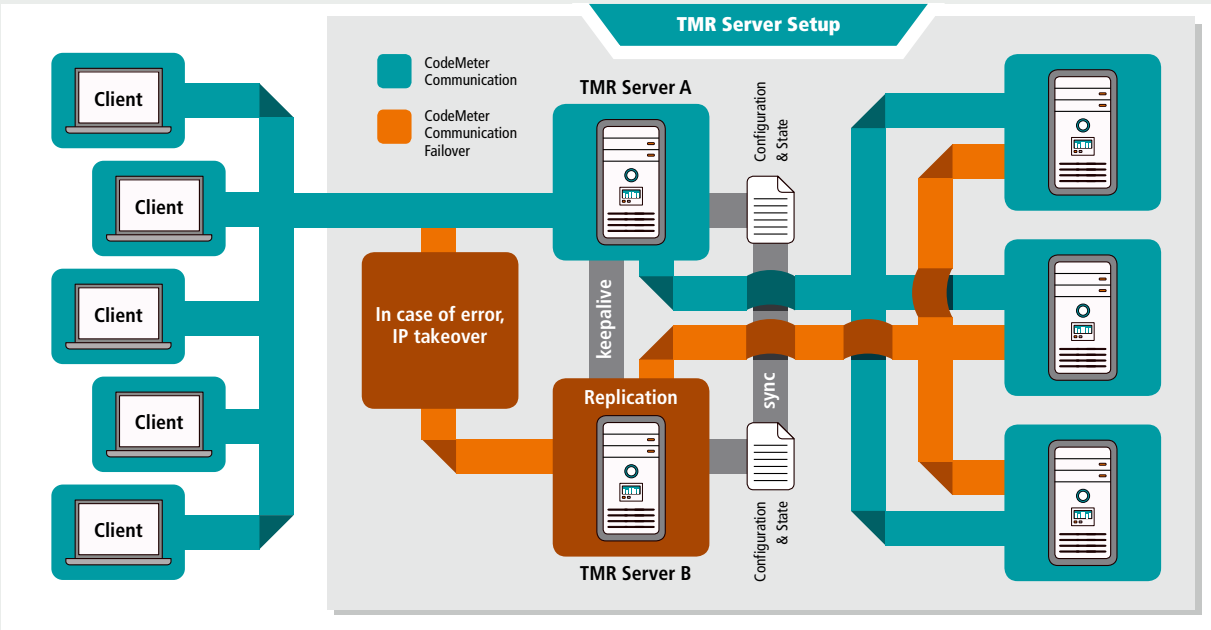
A TMR Server Setup consists of a total of five servers, typically operated as virtual machines. The downstream interface with the clients is provided by a double TMR server that the clients can access via a virtual IP address. The TMR server creates the virtual CmContainers and the TMR licenses within them based on the CmContainers kept on the three upstream CodeMeter servers.

partner immediately of all changes in its state, including the system configuration, the virtual CmContainers, the current license allocations, and all handles used by the upstream CodeMeter servers.

Whenever the servers are forced to switch places, errors or required maintenance interventions can mean a downtime of a few seconds or the potential loss of individual packets.

will facilitate integration into existing business processes. This will include mechanisms for replacing individual CmContainers efficiently, e.g. if some hardware problems require a CodeMeter server to be replaced.

At the same time, the software for the TMR server is expanded to create a license request file for the entire group that contains all the required individual context files. Such update




## Virtual IP Address

In the corporate network, the TMR Server Setup can be reached via a single virtual IP address, which channels all queries to the active TMR server. When the TMR server first connects to the network infrastructure, the switch, it informs the system that it wants to receive packets destined both for its own IP address and for the assigned virtual IP addresses.

The passive TMR server monitors the availability of the active TMR server via a regular keep-alive check. Should the active TMR server not be available, its passive counterpart informs the network that the packets for the virtual IP address should now go to it – seamlessly stepping into the place of the active TMR server.

## Active and Passive

For this feat to work, the passive TMR server has to be aware of the current state of the active server at all times. The active TMR server ensures this by notifying its passive

## Linux TMR Server

The TMR service running on the TMR server is a new development that supports all incoming CodeMeter API calls that are compatible with the first CodeMeter version supporting the Universal Firm Code, namely Version 6.10. The system itself is an original Debian OS equipped with open source software and the TMR service.

Upstream from the double TMR servers, there are the three CodeMeter servers, which can be run under Linux or Windows. With minimal changes to the CodeMeter settings, the licenses on these servers can only be accessed through the two TMR servers. No client could ever use them directly.

## Business Processes

TMR licenses can be programmed separately with the high-level programming API or its incarnation in the command line tool CmBoxPgm. When doing so, the mentioned TMR ID would have to be assigned manually. From early 2019, CodeMeter License Central

files received through the same channel will be unpacked and installed onto the upstream CodeMeter servers in the sequence of their arrival. This ensures perfect cooperation between CodeMeter License Central and the TMR Server Setup – using the same processes that one would use for a single CodeMeter server. ISVs might only have to add tiny adjustments to their activation software.

## Licensing

The first version of the TMR package will be available for licensing from Wibu-Systems beginning in November 2018. Payment is handled by subscription for each installed TMR Server Setup.

# Acquiring Trust with Mini-Certificates

"My installation package includes one application and many libraries. I need to prevent hackers from replacing individual libraries, especially the license.dll. Using checksums might be an option, but I would have to recompile and distribute the entire package again." This would be a typical problem faced by software developers everywhere. Their answer to this challenge, or to similar problems like a secure check of serial numbers, would normally be mini-certificates: a powerful tool for many use cases. What can they do?

## Simple Checksums

A checksum is the end result of a function that turns data strings of any length into a single number. If the data is changed, the checksum changes as well. Popular examples include CRC and Modulo operations, which are frequently used to prevent or reduce mistakes in data entry or transmission. Credit card or IBAN numbers come with checksums to allow the systems to notice incorrect entries before the data is ever sent to the bank, retailer, or service provider.

## Cryptographic Checksums

Simple checksums are not viable defenses against malicious acts. For effective fraud prevention, cryptographic checksums are needed that change substantially in response to even a tiny change to the underlying data. It would be impossible to manipulate the data in such a way that the checksum itself remains unchanged, e.g. by adding blank spaces until the data corresponds to the same checksum again. One popular current choice for cryptographic checksums is SHA256; the MD5 approach that was used frequently in the past is now considered insecure.

Cryptographic checksums are not without their own limits: To test a checksum, one needs the same information from when it was created originally, i.e. the function itself and an optional salt value as the shared secret. This information needs to be kept in the software whose checksums are tested. If even a single attacker manages to get at this data, the entire system is compromised, as valid checksums could be created that could never be identified apart from the originals.

This conceptual restriction has led to some labyrinthine constructs, with checksums for different libraries kept in other libraries again or in the core application – unwittingly creating the problem of updating entire software packages mentioned already.

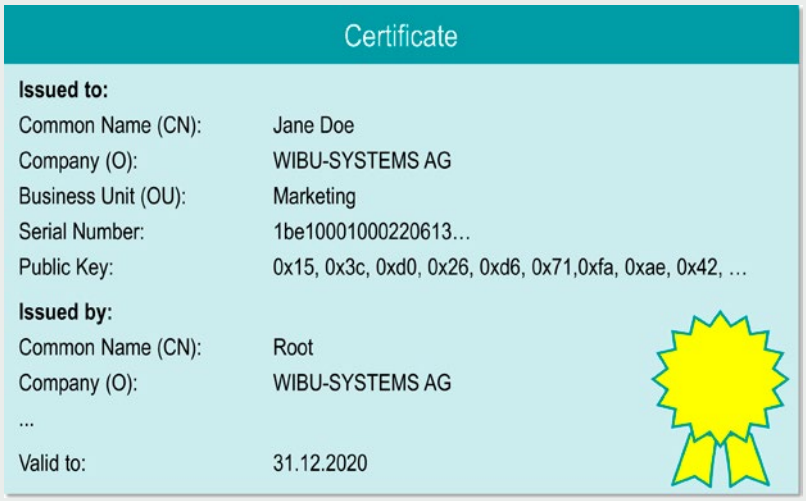
## Asymmetric Cryptography

The solution can come from asymmetric cryptography, in the form of key pairs with one private key that is kept secret and one public key that is out in the open. The private key can be used only by their legitimate owner to sign something; the public key can then be used by anybody to check the signature. Having the public key does not, however, enable anyone to create a valid signature.

Asymmetric cryptography is typically a slow and laborious process that requires data of a certain size, which is why it is often combined with cryptographic checksums. First, a checksum of the data is created and signed with the private key. For later testing, the same checksum is created again and tested against the public key and signature. ECDSA and RSA are established processes in this area.

## The Basic Tools

The basic toolkit for signing our libraries and applications is described here. As the first step in the development process, the public key is included in the software.




After compiling, the private key is used to create a signature for each module (library or application). The signature is delivered alongside the modules, either in a separate file or in a dedicated place in the resource section.

During testing, the checksum is created again, skipping the signature potentially contained in the resources. The resulting checksum is then checked against the signature and the public key.

## One Signature Might Not Be Enough

This could be the end of the story, but the future might bring new challenges that need to be anticipated. Breaking with the clean code principle that calls on developers to not try and solve problems that do not exist yet, security developers should always think two steps ahead. One could ask:

1. Do we want to be protected if the private key is stolen?
2. Do we want to give multiple developers the ability to sign modules?
3. Should modules from business partners be allowed in our product universe?
4. Should test and operational systems be kept completely separate?

These are cases virtually perfect for mini-certificates. A mini-certificate essentially contains one public key, defined sets of rights (flags), and a signature for the public key. They are modeled on the example of the X.509 certificate, but much leaner and easier to use with a binary format defined by you.

## How to Get a Mini-Certificate

In the first step, you create a new key pair, the so-called "root" keys. The public key is again integrated in the software. At the same time, another private key pair is made, and the private root keys are used to create a mini-certificate for the public key of the new pair. The private root key is then locked away, impenetrable and protected from illicit access. Experts recommend a hardcopy and two digital copies kept at two separate places. This is indeed a viable option, because the private root key is rarely needed again after this point.

You then use the new private key to routinely sign the modules you produce. The mini-certificate of the new public key is added to the modules, and the checks compare the certificate with the public root key and the signature of the protected data with the public key on the mini-certificate.

## More Links, Stronger Chains – Chains of Certificates

This approach includes two tiers: The root key pair and the key pair for active use, which would form the recommended minimum set. The system could just as easily include more than two tiers, with the second private key signing the public key of a third key pair, eventually growing into entire cryptographic trees. Depending on the flags you have set, rights can be inherited, e.g. by assigning more mini-certificates.

## Additional Keys

You only need the private root key again – if additional keys are to be created or old keys removed. If the process uses more than two tiers, this is even less likely, because the keys of the second tier can be used for the same purpose. In that case, the root key would only ever be used again if second-tier keys are added or removed.

## Removing Keys

There are two strategies for removing keys:

- A revocation list
- Automatic expiration of certificates

A revocation list would record all void certificates. Devices that cannot access the list (e.g. because they are offline) would still work with the old, void certificates. To prevent this, automatic expiration would enforce certificate renewals. The quintessence would be the required transfer of data to the devices you want to cover. The choice between the two possible strategies should be a choice between "Reliability First" or "Security First".

## Secure Serial Number Checks

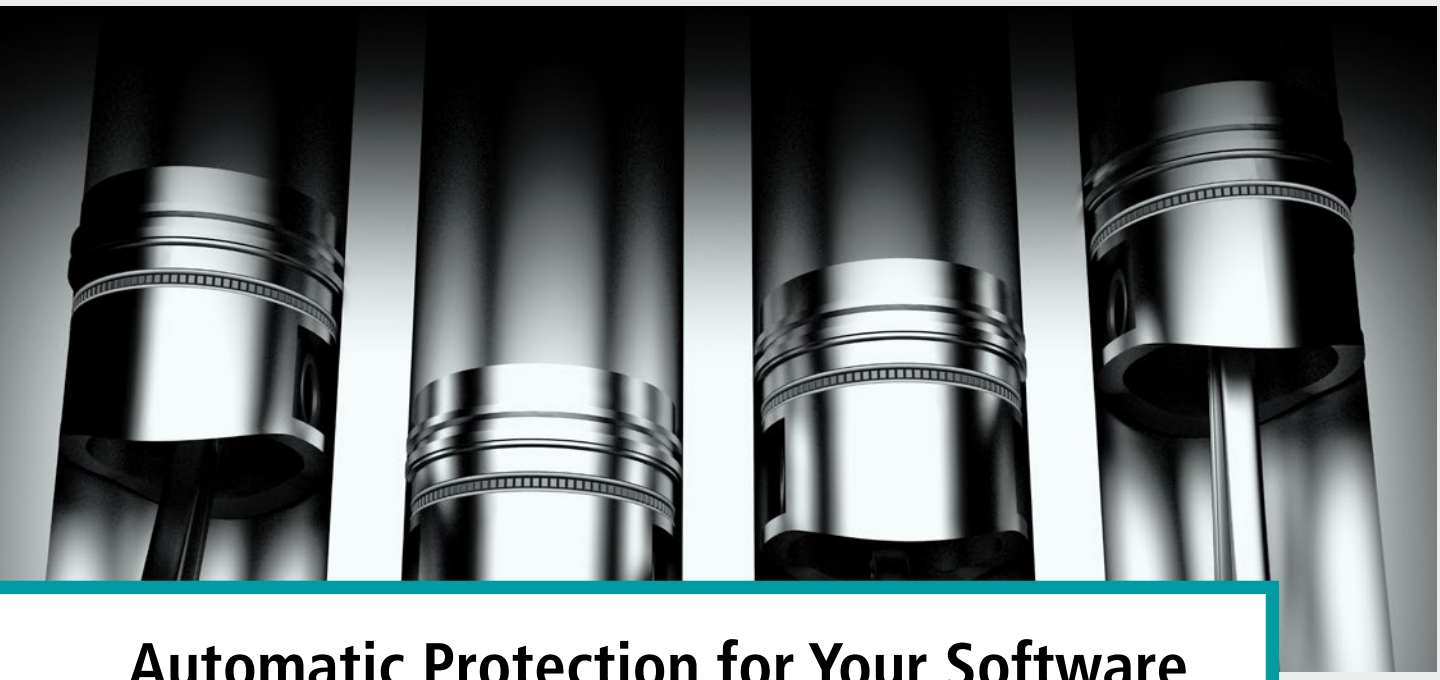
Mini-certificates are also a good choice for checking serial numbers securely: A CmDongle is given a key pair, with the private root key used to create a mini-certificate for the public key. This is then delivered to the user, e.g. as extended protected data on the CmDongle.

The testing system creates a so-called challenge that the CmDongle signs with the private key (or more specifically, both sides each create a part of the challenge). A response for the challenge and the mini-certificate are sent back and checked with the public root key to test the mini-certificate and make sure with the contained key that the response matches the challenge. If all goes well, the identity of the CmDongle is considered proven.

This process is already frequently used to check identities e.g. secure identities for Flexnet or secure authentication of laser machines in service networks.

## Use Cases for Integrity Protections

Beyond acting as a secure proof of ID, the technology is primarily used to safeguard modules from being replaced or otherwise modified. A license.dll could not be replaced with a fake .dll that tricks the device into thinking that a license is available. The system is an elegant and easy-to-use means of encapsulating licenses, but Wibu-Systems still recommends automatically protecting each module with CodeMeter Protection Suite and integrating license checks there. How CodeMeter Protection Suite can bolster the protection for software is outlined in "Automatic Protection for your Software".



# Automatic Protection for Your Software

CodeMeter Protection Suite keeps expanding and growing in strength. AxProtector .NET Standard has joined the Protection Suite family as the tool for .NET Standard 2.0 applications; the other tools continue to evolve and become more powerful. The GUI has been given a facelift, AxProtector .NET is now FIPS-compatible, and AxProtector and IxProtector for native applications now support the execution of code on CmDongles.

## Overview

CodeMeter Protection Suite is a powerful toolbox for the automatic encryption of applications and libraries. It protects executable files from reverse engineering and ties them cryptographically to a valid license.

The individual tools have been tailored specifically to work with each platform or environment to allow optimum protection with minimum effort for you as an ISV. Whichever tool you choose, you select the firm code and product code (or combinations of them) to encrypt compiled libraries or applications. This is done with a key bound to the firm code and product code. Without the right license, decryption is simply impossible.

## AxProtector and IxProtector

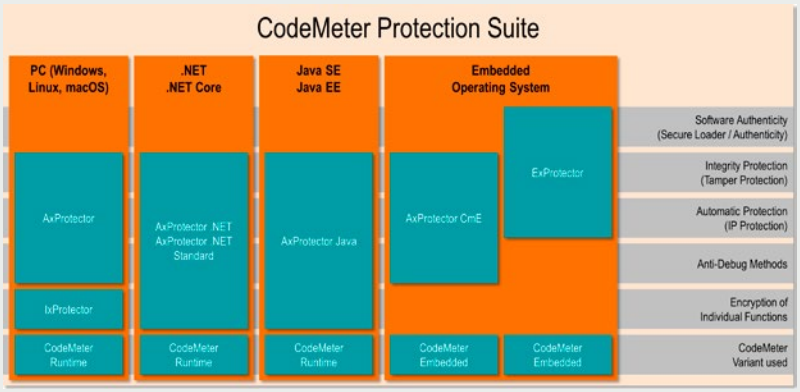
AxProtector protects native Windows, Linux, and macOS applications and libraries. It encrypts the entire code; when the application is launched or a library loaded, the system checks for the required license and decrypts the complete executable code, if the license is there. After that point, the application will perform just as well as if there were no automatic protections at all.

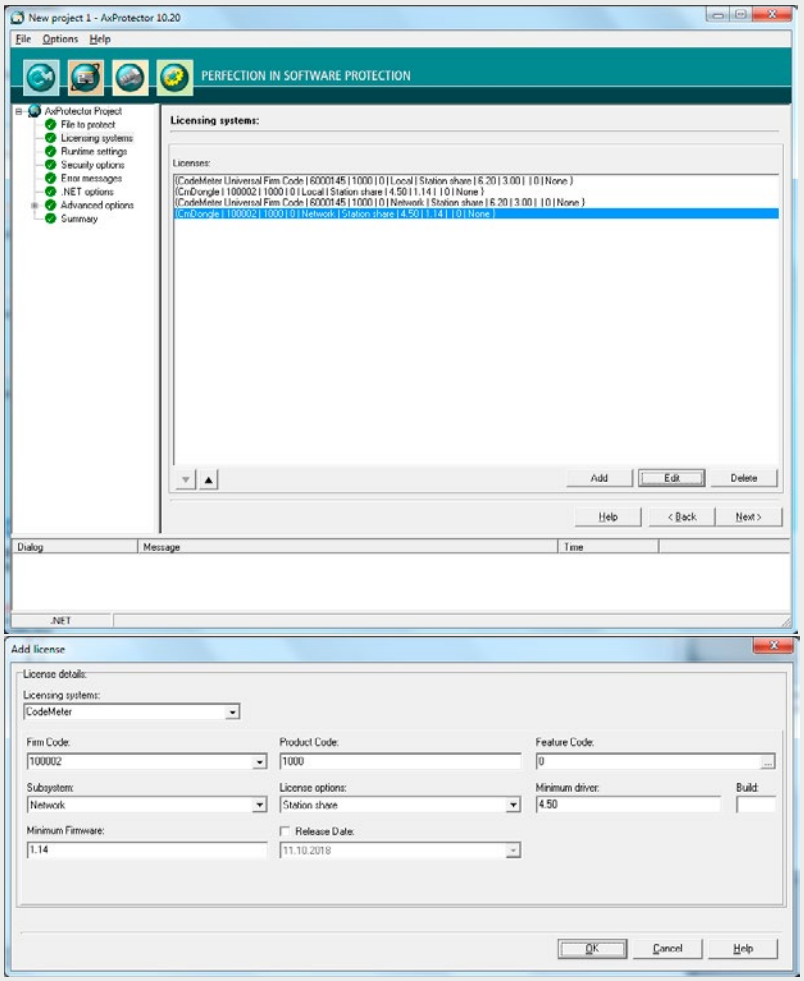
IxProtector allows you to mark and encrypt separate functions to be decrypted and executed during runtime. Depending on the settings, this is either done automatically or by an API call defined by the ISV. It allows you to move particularly sensitive code in separately encrypted form onto a CmDongle to be executed there. At no point is this code visible for would-be hackers.

## AxProtector .NET

AxProtector .NET is CodeMeter Protection Suite's dedicated tool for the automatic protection of .NET assemblies. It does so by encrypting the code of .NET framework applications on the level of individual methods: Each method is given its own stub code that checks for a license when the method is first accessed and only encrypts it if the license is present and correct. This can be compared to how IxProtector protects native code when choosing automatic decryption upon method access.

This approach automatically creates a higher level of security. As the security level has an effect on performance, developers can fine-tune






their chosen approach to find the right balance between performance and obsession for security.

The current version of AxProtector is compatible with FIPS mode, a setting that PC users can configure to ensure that only FIPS-tested implementations of cryptographic functions in the .NET framework are allowed. This is a typically required setting among US official authorities, making it a relevant feature to remember if your users belong to this target group.

## AxProtector .NET Standard

.NET Standard provides specifications for .NET APIs that facilitate interoperability and compatibility between different .NET environments. It makes it easier for ISVs to roll out their applications across platforms or to migrate from one platform to another.

The current release brings the launch of another version – AxProtector .NET Standard – equipped to protect .NET Standard 2.0 applications like .NET Core 2.0 or Mono 5.4 applications.

The protections work on the same basis used with AxProtector .NET: The application is analyzed and protected by encryption on the method level. It can only be decrypted and executed if the correct license is available in a CodeMeter container.

## AxProtector Java

When compiling Java Source Code, the code is first translated into a unique interim language, called Java Byte Code. On the target platform the Java Virtual Machine (JVM) then sees to it that this code is interpreted and executed. This is the secret to Java's platform independence, but it also adds certain weaknesses from the point of view of the security of the Java Byte Code.

As with .NET, the Java Byte Code is simple to decompile and makes reverse engineering less of a challenge for would-be attackers. AxProtector Java has the power to stop them in their tracks by encrypting the code – as its sister implementations do – and tying it to a specific license. The code can be encrypted on the level of classes or methods, and it never touches a hard drive in plaintext form, as it is decrypted on the fly.

Java 9's introduction of modular JARs has led to some changes for AxProtector Java. The new version supports both traditional Java applications created under Java 7 or 8 and modular JARs made with Java 9.

## AxProtector Graphical User Interface

AxProtector shows its true potential with its availability as a command line tool, which enables full integration into automatic build processes as part of continuous integration and continuous delivery. At the same time, a GUI guides users through the features and functions and makes AxProtector more comfortable to use on ISVs' workstations.

The GUI has been given a new facelift, revitalizing, in particular, the settings for licensing systems. Before Universal Firm Codes were introduced in 2016, software developers needed two firm codes to combine software and dongle-based licenses: One for CmDongles and one for CmActLicenses. The Universal Firm Code removes the need for this distinction, and the new facelift reflects this in displaying only one firm code as the standard setting. Developers who still operate multiple firm codes to maintain compatibility are not affected: The option for them needs only one click of the mouse.

## Compatible with WibuKey

CodeMeter Protection Suite supports all CodeMeter licensing systems currently in the field: CmActLicenses and CmDongles. CodeMeter Protection Suite also continues to support WibuKey, the predecessor of CodeMeter originally introduced in 1989. This is what WibuSystems means by long-term availability and lasting compatibility.

## Always Evolving, Continuously Improving

All parts and components of CodeMeter Protection Suite are in continuous development, of course, with particular attention to security and performance. New capabilities like the ability to have executable code on CmDongles or the automatic setting of traps are just two examples of constant evolution. Caching mechanisms have been introduced and refined to improve performance. A committee of experts is charged with overseeing current and planned security features to make sure they deliver real added value and do not impact performance. For CodeMeter Protection Suite, only the best and fastest mechanisms make the cut.


# Progress on the IUNO Research Project

The German national reference project for IT security in Industry 4.0 – or IUNO for short – wanted to understand the threats and risks facing the intelligent factories of tomorrow, develop suitable countermeasures, and put them to the test in powerful use cases. Wibu-Systems and its universal security solution CodeMeter figure prominently in several of the project's work packages and demonstrations. Over the last three years, the members of the projects have worked hard to develop new practical scenarios for Industry 4.0, pinpoint new security needs, and come up with protective solutions in response. This means reconciling different perspectives: Practitioners in the field have very different priorities on the shop floor vs. companies coming from an IT background. The former care most about reliability, while the latter are concerned with the many possible lines of attack the infrastructure is exposed to. IUNO has managed to bring both worlds closer together and empowered people on both sides to see the common ground they share.

## Use Case: Secure Technology Data Marketplace

A technology data marketplace is used to enable trades of data needed in manufacturing processes. Following the lead of smartphone app stores, the marketplace has the potential to make it easier for industry to license and use required designs, parameters, or recipes. The challenge is that the system has to be as seamless and smooth as possible for the intended user, while enabling the licensor to enforce blanket or pay-per-use payment models for their data. At no point should the data be accessible in plain text or usable without the right license. After all, the data is the property and valuable asset of the licensor. The IUNO demonstrator takes the form of an automated cocktail mixer in which case the intellectual property is represented by cocktail recipes. An online marketplace allows consumers to choose the recipe of interest and buy the license for it, pay for it with Bitcoin, transfer the recipe (with the required license and keys) to the mixer, and watch the drink being prepared. Applied to a more industrial setting, a similar system could be used for selling machine settings or blueprints for 3D printers. The system is data-agnostic as it does not matter what type of data is being traded. It can be used across different vendors and systems, covering a wide range of potential customers and environments with a single system.

IUNO
Nationales Referenzprojekt
IT-Sicherheit in Industrie 4.0

## Use Case: Secure OPC UA and RFID Communication

The use case of a secure RFID reader presents a CodeMeter ASIC integrated into the processing unit of an intelligent RFID system made by Balluff. The ASIC provides a space for the secure storage of certificates for OPC UA communication between the RFID reader and its environment. The CodeMeter API is also used to verify the integrity of the data on the RFID tag by checking its signature. Balluff also plans to protect and license individual functions of the RFID reader with the already integrated CodeMeter solution. An ASIC with the CodeMeter Embedded stack can thus fulfill three different functions within the evaluation unit.

## Use Case: TPM 2.0

Working in partnership with Infineon, another work package has integrated the Infineon OPTIGA™ TPM 2.0 with a Linux system to illustrate how software-based CmActLicenses can be tied to an external secure element. The result is a binding scheme that offers a level of protection between a pure software solution and a full-hardware CodeMeter Dongle or ASIC. In environments that already have a secure element in place or where a TPM is mandatory, this allows a more robust binding to hardware traits.

# ALERT

**Do you want to receive more frequent updates from our WIBU world?**

**Subscribe to our KEYflash**

# News in Brief

## CmStick/T

CmStick/T, 1001-03-230, with a real-time clock is now available in a sleek metal case and equipped with all features made possible by the newest firmware 4.10, including Universal Firm Codes. The socketed battery guarantees a long working life.

## CmStick/BMC and CmStick/BMI

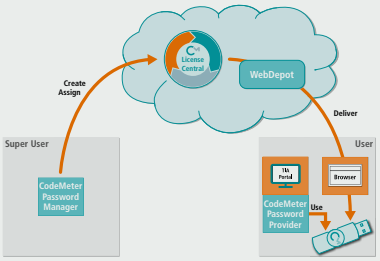
The new CodeMeter sticks with flash memory are now available in an ultra-compact full-metal case with read rates of up to 100 MB/s via the USB 3.1 interface, 16GB to 32GB MLC Nand or 8GB to 16GB pSLC flash memory, and optional support for extended temperature ranges and specialized applications. The comprehensive set of certifications make the CmSticks ready for industrial use.

## Hardware Qualification

CmDongles working in industrial environments, e.g. in controllers, need to fulfill exacting standards. We are investing in the development and testing of our products by external labs for exposure to gasses like $SO_2$, $H_2S$, corrosion, drop shocks, active or passive interference in the form of stray radiation or electrostatic emissions, magnetic fields, humidity, and changes in temperature. All these tests are performed to international standards for CE, FCC, VCCI, KCC, BSMI, RCM, UL and VDE compliance certification – to keep CodeMeter highly reliable for your industrial applications. Contact our QM specialists for more details.

## Siemens TIA Portal

CodeMeter offers secure password management with CmDongles on the TIA Portal, made possible with a special plug-in and management tool for allocating rights via CodeMeter License Central in the cloud. At SPS IPC Drives, we will demonstrate live its use with an HMI panel and a S7 PLC.




## Manager of the Year 2018

Oliver Winzenried was nominated by the editors of Markt&Technik in the Pioneers and Innovators category – and promptly chosen by the readers. A reason to celebrate.


## Big in Japan

To give our Japanese clients the best possible support, we are launching WIBU-SYSTEMS K.K. as a wholly owned subsidiary managed by Tomoki Maruyama and staffed with a top-notch team. Our long-standing partnership with our local distributor will remain active.

## Mono and .NET Core

AxProtector 10.30 now supports .NET Standard 2.0 applications in Mono and .NET Core, including optimum support for Unity 3D.

## Educational Portal for CodeMeter License Central

There is a special portal solution tailored perfectly for educational software. For a demonstration, contact our sales team.

# Success Story | Metrohm

CodeMeter – Protection and Licensing in Chemical Analytics. Intelligent Business Models for Software and Embedded Devices

## The Challenge

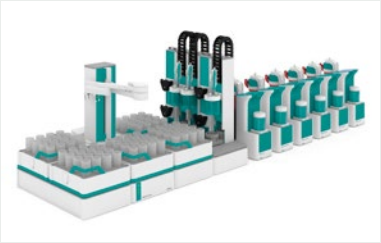
For the launch of its new OMNIS product line, a titration platform for chemical laboratories, Swiss Metrohm AG required a protection system to safeguard the know-how invested in its software and a flexible licensing system that would allow for the creation of customized software licensing models for its laboratory instruments. One specific requirement posed by the company was the necessity to integrate the licensing and entitlement solution seamlessly with the company's existing back office systems.

## The Solution

CodeMeter, the flagship technology from Wibu-Systems, allows Metrohm to protect its software application both on PCs and the embedded operating systems running in its analytical devices and to equip them with the same licensing technology. CodeMeter License Central Internet Edition, Wibu-Systems' server solution for generating and delivering licenses, helps create, manage, and track license activations on both types of systems.

## The Results

By opting to integrate CodeMeter into its OMNIS platform, not only has Metrohm introduced strong protections for its know-how against counterfeiting and reverse engineering, but they can also use the versatile interfaces of CodeMeter License Central to set up a comprehensive portal, activate and track licenses, and monitor license use and activation in the field.



**Bjoern Christensen**
**Head Product Group Software/Workflow/Platforms**
**Chief Systemowner OMNIS, Metrohm AG**

"Every Metrohm product comes invested with three quarters of a century of research and innovation. The expertise going into such groundbreaking products like OMNIS is our greatest asset – which we are protecting with the exceptional licensing capabilities of Wibu-Systems."

Metrohm

## The Client

Metrohm AG is one of the world's leading makers of high-precision chemical analysis instruments. With subsidiaries and sales partners in more than 120 countries worldwide, Metrohm develops solutions for the ion analysis field and has established itself as the global market leader for titration analyzers. In addition to its original operations in Herisau, Switzerland, the Metrohm group also includes Metrohm Applikon and Metrohm Autolab, makers of online analytical devices and instruments for electrochemical research. Instruments for NIR and handheld Raman spectroscopy have completed the Metrohm portfolio since 2013 and 2016, respectively.

## Wibu-Systems Workshops

Wibu-Systems offers you the opportunity to participate in one of the special seminars about:

- Software monetization, back-office integration
- Licensing of software with hardware or software-based keys (SmartBind)
- Code protection against illegal use & reverse engineering
- Solutions for embedded software in systems or cloud applications

Access the latest training schedule by scanning the QR Code or visit **www.wibu.com/tr.**

| Training location | Date | Time |
|---|---|---|
| Paris (FR) | 6 November | 10 am - 3 pm |
| Brussels (BE) | 27 November | 10 am - 3 pm |
| Nieuwegein (NL) | 11 December | 10 am - 3 pm |

**Interested? Please send an e-mail to marketing@wibu.co.uk**

| Contact your local sales representative for details about the workshops and/or upgrading your test kit or current solution to Universal Firm Code with secure offline license transfer and borrowing. | | |
|---|---|---|
| United Kingdom / Ireland | +44 (0)2031474727 | sales@wibu.co.uk |
| Netherlands | +31 (0)747501495 | sales@wibu-systems.nl |
| Spain / Portugal | +34 (0)911230762 | sales@wibu.es |
| Belgium / Luxembourg | +32 (0)28086739 | sales@wibu.be |
| France | +33 (0)186266129 | sales@wibu.fr |

## Wibu-Systems ist auf folgenden Messen und Veranstaltungen vertreten:

**Design, Automation and Embedded**
7 November 2018
Mechelen, Belgium

**Design, Automation and Embedded**
8 November 2018
Eindhoven, Netherlands

**Medica**
12 - 15 November 2018
Dusseldorf, Germany

**Compamed**
12 - 15 November 2018
Dusseldorf, Germany

**Electronica**
13 - 16 November 2018
Munich, Germany

**SPS IPC Drives**
27 - 29 November 2018
Nuremberg, Germany

**IoT Evolution Expo**
29 January - 1 February 2019
Fort Lauderdale, USA

**Embedded World**
26 - 28 February 2019
Nuremberg, Germany

**Imprint**
KEYnote 36
Edition, Fall 2018

**Publisher**
WIBU-SYSTEMS AG
Rueppurrer Strasse 52-54
76137 Karlsruhe, Germany
Tel. +49 721 93172-0
Fax +49 721 93172-22
info@wibu.com
www.wibu.com

**Responsible for the content**
Oliver Winzenried

**Editors**
Marco Blume
Ruediger Kuegler
Wolfgang Voelker
Oliver Winzenried

**Design**
Eugen Olchin

**Print**
Kraft Premium GmbH, Ettlingen, Germany, ISO 14001 certified

Letters are always welcome. We will protect the confidentiality of sources. Third party articles do not necessarily reflect the opinion of the editorial office. Write us at team@wibu.com

Wibu-Systems expressly reserves the right to change its programs or this documentation without prior notice.



Picture credits:
Cover / Page 12
istock.com/3alexd
Page 4
istock.com/NicoElNino
Page 6
istock.com/ijeab
Page 8
istock.com/jakkaje808
Page 10
istock.com/baona

All remaining images are copyrighted by their owner.

SECURITY
LICENSING
PERFECTION IN PROTECTION

WIBU SYSTEMS