FALL 2014

25 years propelling your business to new heights

KEYnote 28

THE WIBU-MAGAZINE

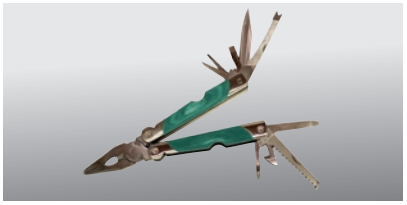
Strong Protection for PC and Embedded Devices

Topics

- Activating Licenses on Embedded Devices
- Protection Suite for PCs
- Protecting the Integrity of Software

WIBU SYSTEMS

Content



Dear Clients and Partners!

The "Internet of Things," intelligent devices, 3D printing, and Industry 4.0 have become much-vaunted and much-hyped concepts. Germany has passed its first IT security law as a "safety belt for IT in critical infrastructure." What all of this has in common is the fact that more and more know-how is invested in production processes, and cyber-physical and embedded systems are increasingly interconnected. Know-how protection, flexible feature activation, tamper-proofing, and cyber-security are becoming essential for the business of machine producers and operators alike.

Copy and know-how protection are the anti-theft devices for your physical machines or construction data. They help stop the tide of global product piracy, whose dramatic scale the VDMA studies have revealed. Technical prevention is becoming increasingly inevitable.

A world that is getting more and more connected, from single sensors to the cloud, offers many previously unimagined opportunities, but also previously unheard-of risks. Cyber attacks can tamper with vital systems, with the would-be attackers at a safe distance. Wibu-Systems provides a unique "airbag" that prevents such manipulation with secure identities, secure boot, and signed code and data at every link in the network. In road traffic, the number of fatalities has gone down not as a result of better roads, but safer cars.

By reading this issue of KEYnote you can learn more about integrity protection, how CodeMeter became a regular part of VxWorks, the real-time operating system of Intel's subsidiary Wind River, and how licenses can be activated on embedded systems. Find out how certificates work, what our Protection Suite can do on your PC, and how custo med uses our solutions to great effect in the medical sector.

Come and visit us at one of our next public events. I would love to get to know you and your business. All that remains for me to say is that I hope this year ends on a high note for you and your business. Wishing you and your families all the best,

Yours, Oliver Winzenried

Benefits in Certificate Usage

Asymmetric signatures rely on fingerprints that are signed with private keys. With the public key of the signatory, anybody can check whether the signature is valid and whether it matches the fingerprint. But where to get the public key of the signatory? And how can we be certain that it is really authentic? Certificates are the answer, especially if there are more possible signatories with unique pairs of keys.

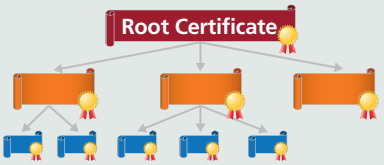
What Are Certificates?

Suppose your company wants to start signing its emails to reassure your business partners that they are receiving genuine emails that have not been tampered with. All of your people are given a pair of keys – but how can all of your business partners receive all of the public keys of all of your employees? It is virtually impossible. The only option is to have a central authority sign the public keys in conjunction with the data of the employees who possess them, i.e. their email addresses and names. The electronic document that stores the public key of the employee and his or her data is called a certificate. All employees sending an email sign it with their private keys and append the certificates. All your business partners now need is the public key of the central authority. They can then verify all of your certificates and – with the public key in the certificate – the email of your employee.

Certificates can contain entire chains of certificates: A third party could authenticate the

Private Key + Certificate

public key of your central authority with another certificate. The certificates of your employees typically include the entire chain.



Certificate Chain

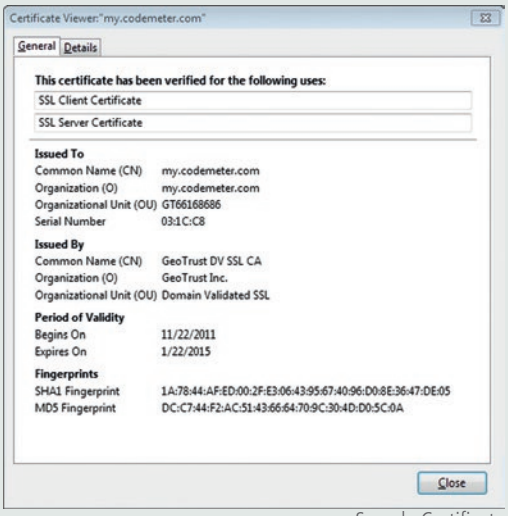
Certificate Authority

There are entire companies specializing in the production of certificates, including the current leaders in the market: Symantec, Comodo, GoDaddy, and GlobalSign. So-called root certificates that contain the public keys of these providers are included in most current operating systems. These providers are called trusted certificate authorities; all certificates created by trusted authorities can be verified immediately.

The special obligation for trusted certificate authorities is to check and inspect all of the individuals and organizations for whom they provide certificates, using such immediate means as contacting the managing directors of their clients directly via a phone number from the public domain (a public phone directory).

Formats of Certificates

A certificate could come in any shape and size. The internet typically uses X.509 certificates, which could be associated with an email address or a DNS name, for example. The application of the certificate can be specified in a dedicated extension (KeyUsage) to allow people to check whether a certificate is actually being used for its intended purpose. A root certificate, for instance, has to have the "certificate sign" key usage.



Sample Certificate

To protect software against tampering, Wibu-Systems uses a proprietary compact certificate format, because X.509 uses lots of resources and is notoriously slow – two qualities that make it a poor choice for small-scale embedded devices in particular. X.509 also lacks some important additional information, although the structure and security features of the Wibu-Systems format resemble X.509.

CodeMeter can store standard X.509 certificates to be compatible with all standard applications, needing only PKI middleware (CSSI) to link up with the standard interfaces (PKCS#11 / Microsoft CSP).

Certificates are not limited to specific encryption algorithms. They can use e.g. MD5, SHA1, or SHA256 as fingerprints and RSA or ECDSA as signing algorithms.

Certificates and Private Keys

In common parlance, we speak of "signing something with a certificate." This is incorrect – we sign something with a private key and add the public key, authenticated by the certificate. The private key is not part of the certificate, although there are certain file formats in which the certificate and the allocated private key are kept in a single file.

CodeMeter offers a means of storing private keys securely in a CmContainer. The private key remains in the container, while the fingerprint is sent by the CodeMeter API to the container to be signed. CmDongles use their built-in smart card chip for the purpose, while "soft" CmActLicenses use the CodeMeter Runtime, which operates as a service or demon and is protected against illicit accessing or debugging. The certificate itself can be stored in the CmContainer, in a readable component, or as a file on the hard drive.

What Happens to Lost Certificates

Losing a certificate itself does not cause any problem, since the certificate includes no confidential information. Losing a private key for a certificate (or allowing somebody else to acquire it), on the other hand, can have dramatic consequences. A person with the private key can sign anything in the name of the original owner and use the certificate to authenticate the corresponding public key.

This was one of the reasons for the impact of Stuxnet. Software developers sign their code, especially for drivers, with their private keys. Certificates from trusted certificate authorities are used to confirm the public keys. Virus scanners typically work with points systems; software from reputable sources gets bonus points. In the Stuxnet case, the private keys of two known producers of drivers were stolen and used to sign the virus. This meant that the virus could not be recognized by anti-virus software and had an opportunity to spread undetected.

For such cases, there are so-called certificate revocation lists (CRL) with the immediately identifiable serial numbers of revoked certificates. These CRL are themselves signed to ensure their integrity.

Both certificates and revocation lists have distinct expiry dates, making it necessary to update both of them on a regular basis. This is not a problem for internet-connected PCs, but it is an issue for embedded devices that are expected to control hardware over many years without ever going online. It should be carefully considered for how long a certificate should stay valid: If a beverage maker's plant stops operating, it would simply not produce any profit. A blast furnace, however, would become unusable for good.

How to Obtain Certificates

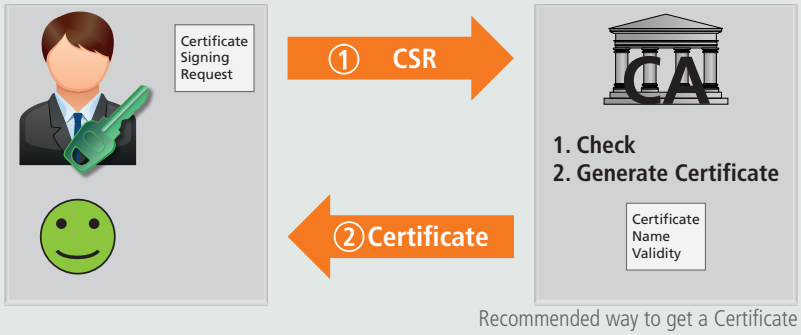
There are two basic ways of obtaining a certificate. The traditional way is by personally creating a pair of keys. CodeMeter can do so either by means of the CSSI middleware (pair of RSA keys) or by means of the CodeMeter API (RSA or ECC keys). When using a CmDongle and a pair of ECC keys, this can also rely on the random number generator integrated on the smartcard chip. The private key is created on the dongle and never leaves it. The public key can be calculated by the CodeMeter API.

In the next step, a certificate signing request (CSR) is produced, which already includes relevant data like the owner's name and the public key. This CSR is sent to the certificate authority.

The private key of the certificate authority is used to produce a certificate, normally by checking and copying data from the CSR. Other data or data fields can be added if required. If the certificate authority is not a root certificate authority, its certificate is also included in the client's certificate.

The finished certificate is sent back for storage by the client.

In practice, this can be a complicated effort. Many people choose an alternative approach: The certificate authority creates the pair of keys, then produces the certificate, and finally sends the certificate and the private key to the intended owner. This simpler procedure has distinct drawbacks:



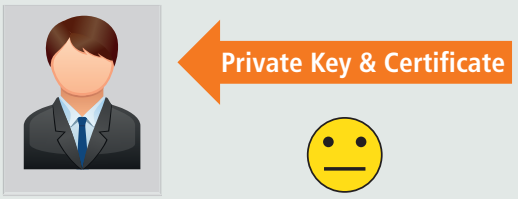
Recommended way to get a Certificate

Two parties – the owner of the certificate and its issuer – now know the private key. The intended use and the potential risks determine how grave a problem this would be. An operator of a website wanting to identify his users by way of a self-created client certificate will not have a problem with the method. Another disadvantage is the way in which the private key is transferred, which should be protected from the prying eyes of unauthorized third parties. This normally means password protection.

CodeMeter offers a simple means of securely transferring private keys. A license (which can contain keys in the data) can be placed on a CmContainer by remote programming. The user first creates a remote context file (WibuCmRaC) with the public key of the chosen CmContainer. The creator of the license uses this to create a remote update file (WibuCmRaU). The license data in that file is encrypted and can be decrypted only by the right CmContainer. Decryption takes place within the container itself, so that the private key is never present in clear text form outside of this secure environment. CodeMeter makes creating and distributing private keys and certificates easy.

Self-Signed Certificates

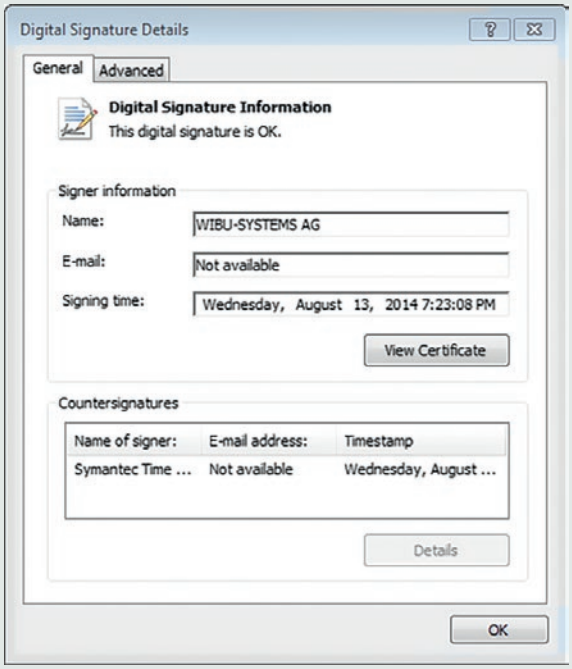
A self-signed certificate is a type of certificate that one creates by means of one's private key. It is a simple way of creating certificates, especially for trial purposes. For a self-signed certificate to be accepted, it normally has to be added manually to the list of trusted certificates. This typically limits their use to testing and trials.



Typical Use Cases

A typical area of application for certificates is their use as **server certificates**, which establishes the identity of a web server and encrypts the communication between end users' browsers and that server.

Client certificates are another form. The operator of a website creates and distributes certificates, that browsers use in combination with the private key to access the server. The server can then read the certificate and award specific access rights, e.g. for named organizational units (OU).



Code signature

Code signing certificates are a third type of application, with software developers signing their code, whose genuine origin is verified by means of a correct certificate. Windows' built-in system was originally designed to protect end users from malware and viruses. CodeMeter now uses the same idea to protect the integrity of entire embedded devices.

Easy way to get a Certificate

Other use cases include the encryption of email messages or the new German identity card (NPa).

OPC UA

OPC UA (OLE for Process Control Unified Architecture) also relies on certificates, which are used as server and client certificates. CodeMeter again makes easy work of creating and managing these certificates. Finished certificates can be distributed simply by plugging the right CmDongle into the controller or PC.

Conclusions

Certificates are an essential tool for identifying individuals or devices and for verifying whether data is genuine. Certificates and the private keys allocated to them can be used to establish truly secure lines of communication. They are an ideal addition to licensing and copy protection systems like CodeMeter. CodeMeter has been designed to work hand-in-hand with certificates and private keys, which are stored in a secure and unreadable format in CmContainers. CodeMeter supports the X.509 standard as well as a lean proprietary format for systems with limited resources.

Protecting the Integrity of Software

A common use case of software certificates is code signing: the developer of a software application signs the code with a private key. A certificate is produced by a trusted certification body and links the public key with the identity of the software developer. This allows the end user to verify which developer made the software he or she is using and whether it has been tampered with. This common mechanism was originally developed to protect end users from the all too frequent threat of viruses. It is not, however, enough to give the software developers themselves a means of avoiding piracy or tampering. This is where AxProtector and ExProtector enter the fray.

Windows uses a built-in code signing mechanism (Authenticode) to notify users when they are using software from an unknown source, that is, software that is not signed, whose certificate cannot be traced back to a trusted root certificate, or whose signature is incorrect. However, the users are only giving a simple warning message, and they can even opt out of these messages. Little reason for the makers or users of pirated software to stop their wrongdoings.

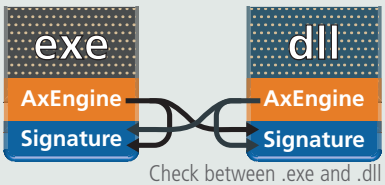
Windows - AxProtector

AxProtector encrypts the entire executable file (.exe) and appends a fingerprint with a dedicated signature. The public key is also hidden within the application.

Trust no one, but yourself – that is the thinking behind this approach, which has the protected application check itself upon launching. If its signature does not match the application in the RAM, the application will close and notify the user. With AxProtector, the application trusts itself alone, but that is not the only step that goes beyond the standard protections built into Windows. The protected application checks itself in the active RAM, and not simply the files stored on the hard drive. This means that any changes introduced by hackers after launching are also identified.

Applications and Libraries

One could wonder whether a hidden fingerprint (checksum) might not be enough and deliver the same results as such a check of a built-in signature. This might be true of applications that consist of single .exe files. The main advantage of the system is particularly evident when the application in question consists of an .exe file and several libraries (.dll).



Check between .exe and .dll

In such cases, AxProtector signs the .exe itself and all related .dll libraries. When a protected library is loaded, the .exe verifies the integrity of the .dll, and is in turn verified by the .dll. As the developer, you can decide whether the .dll can be accessed only by a specific (your own) application. And it is up to you to say

which .dlls need to have a valid signature to be accessed.

This opens up new possibilities for other use cases and for other threat scenarios:

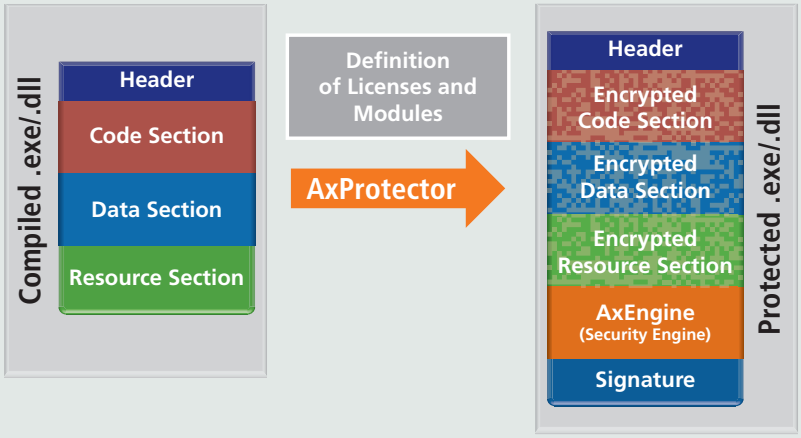
- A hacker might create a changed copy of one of your .dlls. He or she might replace the .dll that tests your application's copy protection with a .dll that always reports a positive response. AxProtector prevents such sleight-of-hand swapping, making the hacker's work considerably harder. Relocating copy protection to an external library in this manner is, however, never a recommended option.
- A hacker might use a library with your intellectual property for his or her own application. The integrity check instructs the .dll that it is not being used for the application you have authorized, and it simply stops working.

Why use signatures? Answering this question recalls the chicken-and-egg problem: For the .exe to check the .dll, it would have to know its checksum, which needs to be stored in the .exe file. But for the .dll to check the .exe in turn, it would also need to include the .exe's checksum, which would change the .dll and void the original checksum in the .exe file.

Using signatures means that the public key alone needs to be built into all modules (.exe and all .dll files). A private key is used to sign each individual module, and the signature is appended to each module. Every piece can therefore check every other piece, and individual pieces in the puzzle can even be updated without affecting the other pieces.

ExProtector – Embedded Devices

AxProtector produces protected applications or libraries that decrypt themselves. ExProtector takes this one step further and protects applications, libraries, or even entire operating systems on an embedded device. To launch or execute any protected element, a specific module – the ExEngine – is first integrated in the executing piece. If the entire operating system image (e.g. a VIP in VxWorks) is encrypted, the ExEngine is integrated in the bootloader itself. If only individual applications or libraries (e.g. RTPs or DKMs in VxWorks) are encrypted, the ExEngine operates from within the operating system.



AxProtector: Protection Process

When setting up the protection, you can determine whether the protected software is encrypted, signed, or signed and encrypted. Encryption offers protection against reverse engineering; signatures guarantee the software's integrity.

As in the case of AxProtector, the encryption uses a key from the license when AES-encrypting the files. The signature uses a private key to sign the fingerprint in the application. Both the AES key and the signature key are stored securely in a master dongle (the Firm Security Box – FSB). Both keys are safe against unauthorized access or illicit use.

Managing Permissions

Simple set-ups include single or very few master dongles, each containing the same private key. The related public key is integrated in the ExEngine when developing the software for the embedded device.

ExProtector also helps with managing permissions. For this purpose, each master dongle is given its own private key for the code signature. Certificates are produced by a central actor to define the permissions for each master dongle, which can distinguish between different application levels or types and batches of devices.

The number of application levels is fixed already when integrating the ExEngine in the embedded device. In a simple scenario, this would distinguish between operating systems, applications, and settings. Device types and batches can also be defined individually.

This system allows the producer of embedded devices to create certificates with which the end users can update applications on their devices themselves, while updating the operating system as a whole remains the preserve of the original developer. The user can only load operating system images that the original developer has signed. The system also allows the differentiated and sophisticated handling of the permissions even at the original developer himself: for instance, Production alone could have the right to sign software for production devices, whereas developers could only sign trial devices. CodeMeter offers you a signing tool for the simple creation and administration of the certificates you need.

Conclusions

Whether for embedded devices or PCs with run-of-the-mill operating systems, AxProtector and ExProtector protect your software reliably with encryption against reverse engineering and with signed code against later tampering. Used on standard operating systems, AxProtector is the living embodiment of the principle "Trust no one, but yourself," and offers unparalleled protection. On embedded devices, ExProtector offers the additional option of using certificates to manage permissions and define once and for all "who can update what and where."

CodeMeter Security and VxWorks 7

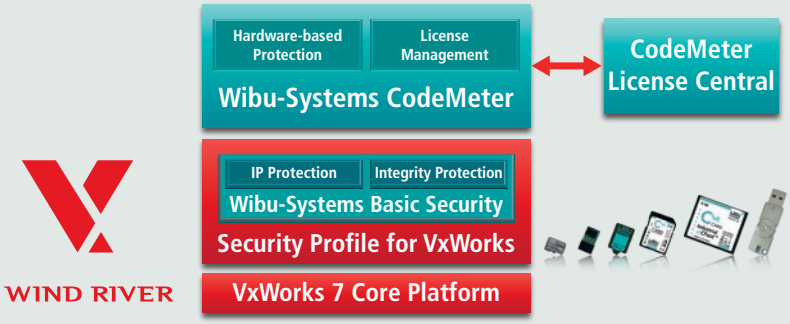
Powering more than 1.5 billion embedded devices, VxWorks is the world's most popular real-time operating system. The users of VxWorks are increasingly interested in security measures that are quick and simple to integrate. CodeMeter technology is compatible with the VxWorks development environment and the operating system itself. With VxWorks 7, using modern security protection technology is even easier.

The constant stream of news about security exploits and industrial espionage is powering a new demand for embedded systems that are designed to be inherently secure without relying on external protection systems like firewalls or VPNs. Mechanical engineers would call such devices intrinsically secure. Devices without significant security capabilities will find fewer and fewer buyers in the foreseeable future. At the same time, the developers of applications that run on embedded systems want to protect their intellectual property (IP). The security solutions should allow maximum protection with minimum effort. After all, not every user is also an expert cryptographer. The needs of both target groups – the developers and plant engineers, and the users and operators – were considered in the design of the new Security Profile for VxWorks 7.

To make it easier for end users to work with cryptographically protected software and secure boot procedures, Wind River has teamed up with Wibu-Systems to include Wibu-Systems' technology in Security Profile for VxWorks. The profile is being sold by Wind River and can be used as a plug-in for developers' workbenches. In addition to Wind River-developed features, it includes tried and tested components from Wibu-Systems that have been part of VxWorks since version 6.8. The operating system image, the kernel modules, and the applications are still encrypted by ExProtector. ExProtector and the CodeMeter Embedded driver (now as Version 1.7) are both part of Security Profile package.

The difference is that Security Profile works without CodeMeter Dongles or computer-specific licenses. The protection is purely software-based, but embedded deep in the VxWorks kernel. The solution therefore complies with two essential security requirements: integrity and know-how protection. The integrity of the individual software components is protected by cryptographic signatures. The VxWorks development environment includes its own certification authority (CA) that produces, signs, and manages the required certificates. The software vendor can provide a certificate for every developer involved in the project, which identifies the developer and determines his or her permissions. Even in large-scale projects, this makes sure that only named developers have the right to modify kernel modules or generate new VxWorks images. Every developer signs off his or her work with a personal certificate. When the finished software is run on an embedded system, the Secure ELF (Executable and Linkable Format) loader checks the chain of certificates immediately in the operating system to establish whether the signatures are valid. If this is not the case, the application will not run.

For our non-IT specialist readers: If a single bit or parameter is changed – for whatever reason – in a signed application, the signature is automatically voided. Checking the signature makes sure that the application has not been tampered with and that it comes from an authorized developer who alone has the right key.



Signatures protect applications from tampering and make sure they are from an authorized source. In order to avoid the theft of intellectual property by means of reverse engineering, software developers also need to encrypt their code. This is also possible with Security Profile: when setting up a new VxWorks project, an AES key is created for encrypting all modules and applications. The files protected in this manner are distributed in encrypted form only, while the right keys are kept at both the software vendor's and on the embedded systems. The Secure ELF loader decrypts the files in the operating system only when an application is launched. The necessary function is integrated in VxWorks itself and needs no adjustments on the part of the developers.

Secure Boot

Developers or plant engineers want to make sure that their machines controls only use software they have tested and approved and that the controls cannot be tampered with. This level of protection is already possible for the software itself in the form of code signatures. Making sure that the operating system, i.e. VxWorks, itself has not been manipulated needs a secure boot function, which was previously discussed in KEYnote issue 26. Platforms that support UEFI (the successor to the former BIOS) can make sure that only approved and signed software is run from the very first booting to the launching of individual applications. A key function of UEFI is its support for secure booting: the bootloader itself is checked, which launches only signed firmware images to run signed applications only. UEFI is the secure anchor that holds the entire secure boot chain in place.

CodeMeter Security

Security Profile is fully compatible with CodeMeter Security. Keys can be stored in hardware dongles (CmDongles) or system-specific soft licenses (CmActLicenses). In addition to the secure storage of the keys, CodeMeter also adds copy protection, as neither a CmDongle nor a CmActLicense can be copied.

In addition, CodeMeter Security allows the use of flexible licensing models and works with CodeMeter License Central to create and issue licenses. This paves the way for novel business models for embedded devices, such as the leasing of production equipment, pay-per-use concepts, or the secure monitoring of allowed production runs and batch sizes. CodeMeter Security also helps emulate the success of the after-sales business in the overall consumer industry and in the smartphone industry, which is experiencing particularly rapid growth.

Devices can be delivered with all features ready for use, but the client is limited to the features that his or her license covers. All other features remain off-limits until the right license has been bought. This saves considerable effort for product development, testing, and certification.

Security Profile offers a comfortable entry point for embedded security with cryptographic technology based on AES and elliptic curves. Secure Boot makes sure that applications can be run in a trusted environment.

CodeMeter Security is the option of choice for users wanting to add flexible license management, copy protection, and high security standards to their keystores. CodeMeter Security is based on established mechanisms and is distributed by Wibu-Systems as an add-on for development environments.

Complementarity of Security Profile and CodeMeter Security	
Integrity	✓
Authenticity	✓
IP Protection	✓
Certificates	✓
Copy Protection	optional CodeMeter Security
License Management	optional CodeMeter Security
Hardware Key Storage Containers	optional CodeMeter Security

Activating Licenses on Embedded Devices

Industry 4.0 and the Internet of Things are a vision of all embedded devices being interconnected in the future. In this future reality, the security that our devices ensure is key: when everything is connected with everything else, entirely new forms of threats will arise. Hackers could hijack trains from the safety and comfort of their living rooms or even sabotage the power supply of entire countries. CodeMeter's® encryption and identity / permission checking mechanisms provide the backbone for the right response.

Apart from security concerns, the ability to unlock features on demand is becoming increasingly sought-after. However, different devices might operate with similar hardware. Their features and price only differ as a result of software settings or additional software modules loaded onto the devices. CodeMeter goes beyond security functions and offers a complete system for software protection and license management. But how would a license reach an embedded device?

A Connected World

The connected world of the future needs no complicated solutions: the manufacturer of the device simply provides a CmContainer alongside it, which can be empty or already equipped with activated licenses. Hardware dongles like the CmDongle are simply hooked up to the device in the form of CF, SD, or uSD cards or USB dongles. Purely software-based solutions have CmActLicenses integrated in the system by means of the CodeMeter API.

To activate a license, the manufacturer creates a ticket in the CodeMeter License Central (coming in the form of a sequence of characters: NFGSX-VWNYJ-T74CD-48H5B-7NEEJ) via SAP or a similar ERP system. The relevant licenses are married to the ticket and stored for retrieval in the CodeMeter License Central, which is hosted either by the device manufacturer or by Wibu-Systems. The end user receives his or her ticket by mail or with a physical delivery slip.

Once the end user has received the ticket, he or she enters it on the embedded device as planned by its manufacturer. A remote context file is created by the CmContainer, which includes the serial number of the dongle or a fingerprint of the device in the case of soft licenses. The ticket and the remote context file are transmitted to the CodeMeter License Central.

The CodeMeter License Central checks whether the ticket is still valid and whether it has not been used before. If the answer is positive, a remote update file is created with the waiting licenses. CodeMeter uses cryptographic means to make sure that this remote update file can only be placed in the CmContainer it was meant for. The network sends the remote update file back to the embedded device. Creating and uploading the remote context and update files is done with CodeMeter's own API functions.

If the CodeMeter License Central is not used, the remote update file can also be created by means of the CodeMeter API or a command line tool. This option is meant in particular for trial and integration scenarios.

The Offline World

The remote context and remote update files can also be transported by offline means. For the purpose, the maker of the embedded device integrates the necessary processes in a PC-based development or support tool, which can be hooked up to the embedded device. The embedded device is prompted to create a remote context file (or this file is created automatically in the background), which is then transferred to the PC. The PC connection means that this route is possible even if the embedded device has no display or entry devices of its own.

The PC tool then contacts the CodeMeter License Central at a later point, even after the connection with the embedded device has been removed. The tool requests the ticket from the end user, before sending the ticket and the remote context file to the CodeMeter License Central and receiving the remote update file in return.

In the next step, the PC is reconnected to the embedded device, with no internet access required anymore. The PC tool transfers the remote update file to the embedded device, either initiating the update itself or relying on the embedded device scanning regularly for updates and launching them automatically in the background.

If an internet connection is available, the remote context file can also be transferred from the embedded device to the PC and sent as a receipt to the CodeMeter License Central.

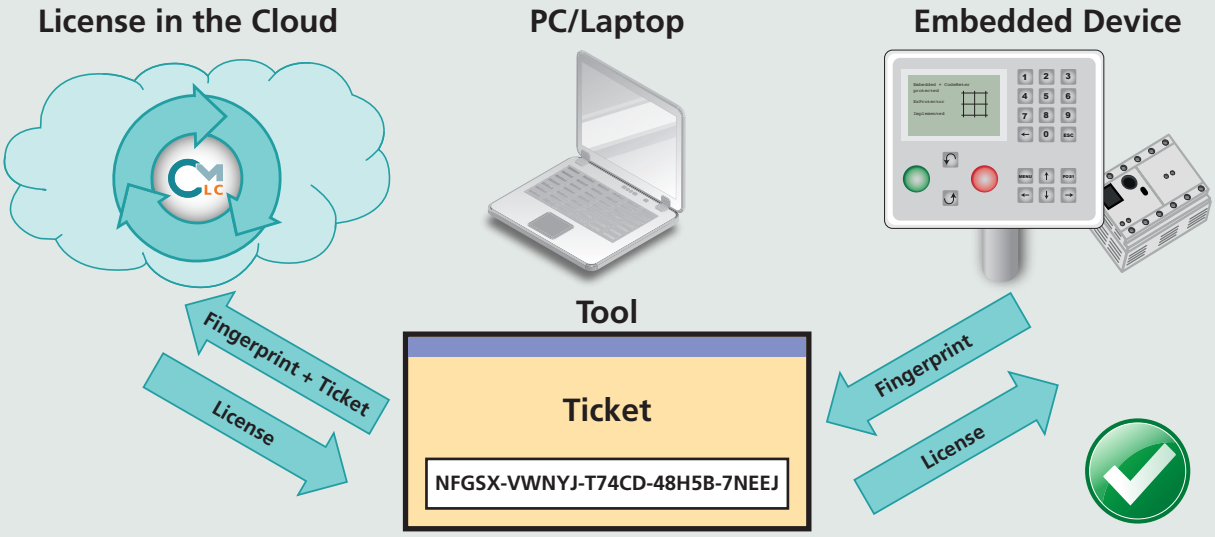
Offline without Uplink Channel

In certain use cases, the maker of embedded devices prohibits the establishment of uplink connections. In such instances, the remote context file can be simulated in the CodeMeter License Central by way of the serial number, fingerprint, original remote context file, and completed updates. This simplifies the process: The user starts the PC tool, enters the ticket, selects the serial number of the embedded device, and downloads the remote update file. The rest of the process proceeds as described above.

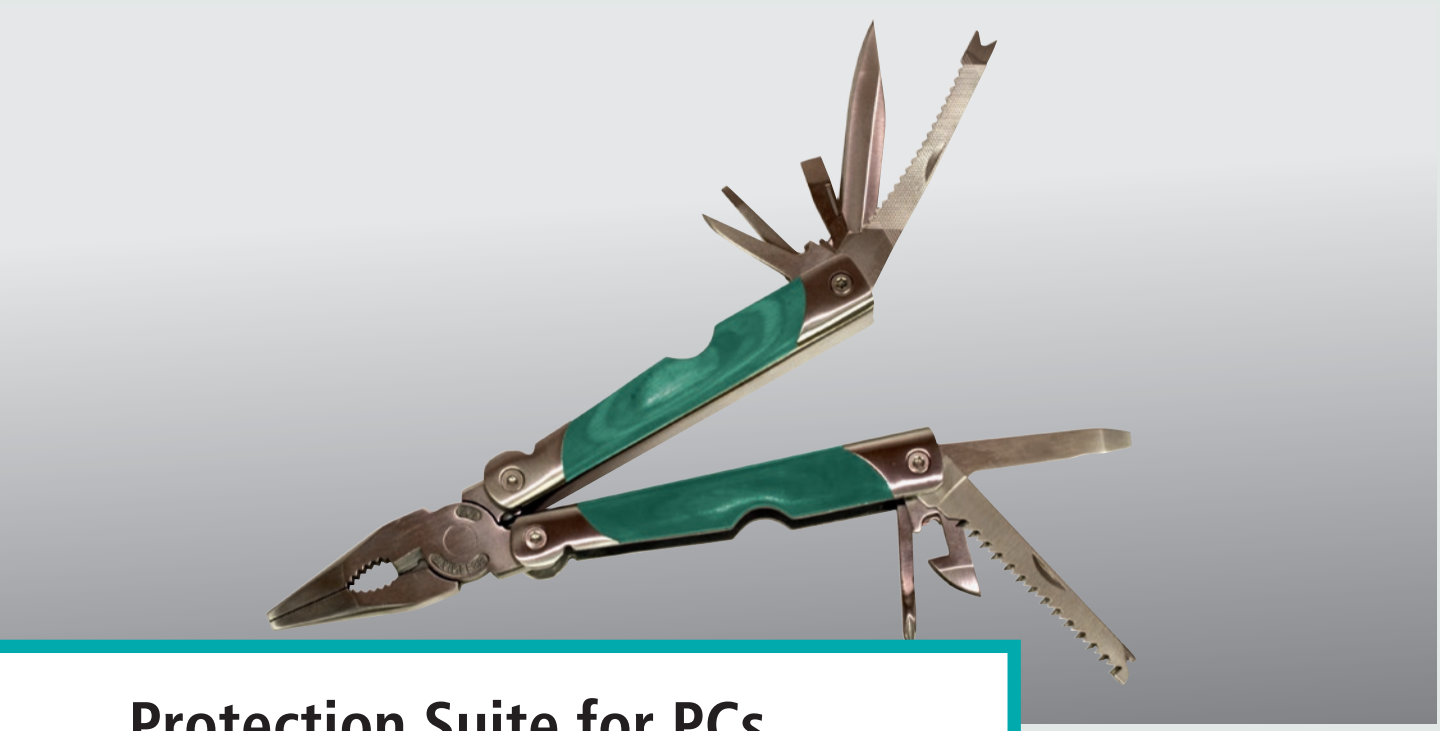
In these cases, the maker of the embedded device cannot ascertain which licenses were actually activated on the device in question. If a fingerprint changes, e.g. because the hardware has changed or a new CmDongle is being used, a new remote context file is required.

Conclusions

Activating a license on embedded devices via CodeMeter uses the same process as online activation on PCs. A ticket is used to authorize the right licenses for retrieval. The ticket is a 25-character code created by the device's manufacturer. In offline scenarios, a PC can be used as a bridge between the embedded device and CodeMeter License Central. The request is transferred as a Remote Context file, and the license as a Remote Update file.



License Activation on an Embedded Device



Protection Suite for PCs

The many versions of the AxProtector technology for different software application types offer the right protection for your specific needs. As a software developer, you want to protect your revenue stream and safeguard your intellectual property.

The Wibu-Systems Protection Suite is a comprehensive set of protection mechanisms for software products. The need for protection has become ubiquitous as the threats have multiplied. Payroll accounting applications in a catering business, music editing software in a recording studio, or digital grinding machines in a dental laboratory: they all need protection.

There are many different threats that independent software vendors need to worry about. Every single piece of software that is sold should be used for its intended purpose, e.g. a single-user license for a basic set of features is meant to be used to this exact scope without the possibility to activate non-purchased additional modules. Licenses are the solution, but they need to be secure. License controls need to be built into the software in a tamper-proof, non-removable manner if you want to avoid the internet being flooded with instructions for circumventing your license protection or even torrents of pirated copies of your software.

More and more companies today need to inject a lot of their technical know-how into the products they sell. This can be, for example, the most effective way to place individual pieces on a cutting pattern, or the secret code that allows one video editing application to process each frame more quickly than its competitors. Such intellectual property needs to be safe from the prying eyes of competitors at home and abroad. Keeping the head start for as long as possible translates into real financial gains.

It is becoming increasingly common for features to not be sold on a blanket basis, but for licenses to be limited to the required modules, a specific time period, or to a defined number of uses for specific features. This even allows users on a limited budget to use the software they need – a win-win solution for license vendors and their clients.

The means of protection need to cover a range of mechanisms:

- **IP Protection**: Automatic protection means encrypting the binary code and making it unusable for reverse engineering tools. Integrated methods to recognize such attempts while the code is running ensure that the protection cannot be undermined on the go.
- **Integrity protection**: Checking code upon launching and while running reveals whether the code has been tampered with before or during its use. If necessary, the running application can be terminated immediately. Changes to an executable file with which attackers snoop on users or deactivate built-in protection functions are easily detected.
- **Increased security for modules**: The additional encryption of individual methods gives extra protection to the intellectual property contained in them. Without the right key, no attacker can get at this know-how, and even with the right key, the information is only kept in unencrypted form for a very limited period of time.
- **Individual licensing models**: The API includes methods for your individual models, be it your payment models for pay-per-use applications, the use of information contained in the license, or other individual protection mechanisms. This gives you the right options for all requirements not covered by the automated mechanisms.

- **Software authorization**: Limiting software to systems that were authorized for it is one of the most important requirements, in particular when working with controller devices or other embedded systems.

Distinctions

Different software application types need different types of protection. This is why AxProtector is the umbrella term for a versatile kit of tools for all application types. ExProtector, the dedicated protection system for embedded systems, was showcased in detail in the last KEYnote (No. 27, page 8).

Native Applications

AxProtector guards native applications written in C++, Delphi, Cocoa, or other languages on Windows, Mac OS X, and Linux. Large parts of the binary code are encrypted, loops are realigned or replaced, security and library functions added. The resulting executable file or library is unusable for attackers. It needs the right license to be used.

Additional encryption of individually defined functions, which have to be decrypted by specific API command, adds another degree of security. Called IxProtector, this method also allows you to integrate traps as a perfect weapon against would-be attackers.

.NET Assembly

.NET applications consist of precompiled code that is particularly easy to decompile. AxProtector .NET extracts the contents of all protected methods and stores them as encrypted data. For the run-time, this data can be automatically decrypted and provided dynamically at the point of need. The encrypted .NET Assembly remains valid .NET code, but only includes the code bodies. Coupled with the integrated obfuscation of private and internal methods, these pieces allow very little meaningful insights into the actual functions.

Java

The situation is similar in the case of Java. There are many tools to decompile the compiled code. A particular challenge is posed by the built-in debug interfaces and the option of rebuilding the Java Virtual Machine itself. AxProtector Java can encrypt the stated classes, which are decrypted as native code for the run-time and made available for the Java Virtual Machine.

Modern application servers like GlassFish or WebSphere in particular need classes to be loaded piece by piece for the available optimizations to work. IxProtector allows you to keep the class itself unencrypted, but to encrypt single, multiple, or all methods within the class, with simple settings set as annotations in the source code. The methods are then encrypted automatically for the run-time.

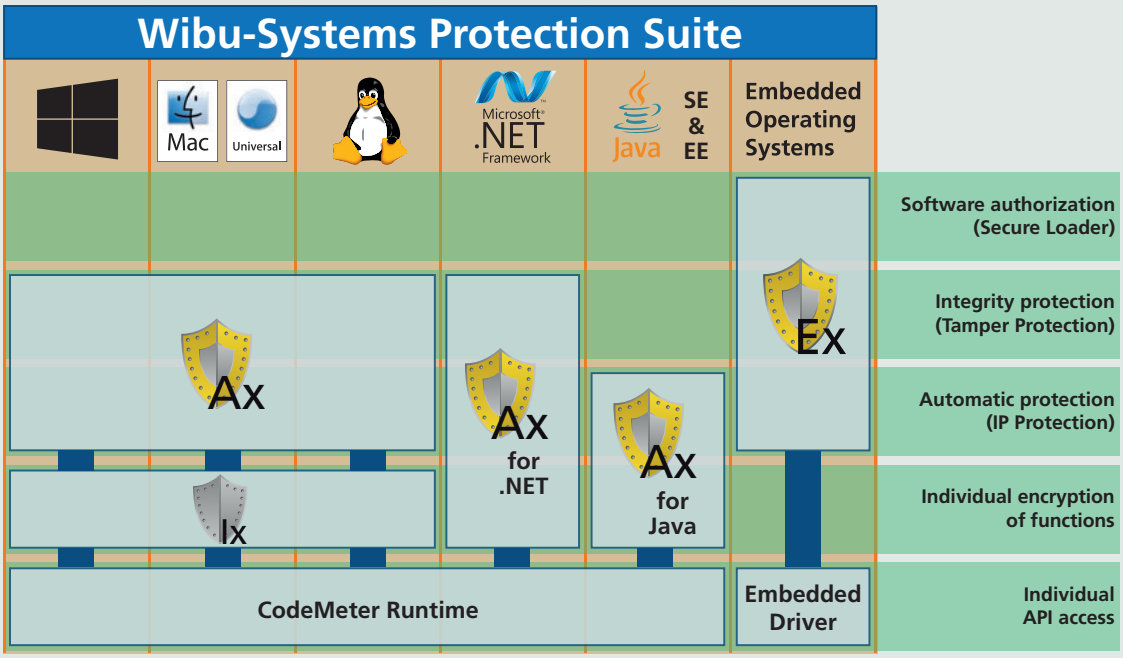
But wait, there's more…

The AxProtectors for the different types of applications offer a range of appealing additional options. Be it the automatic blocking of licenses after attacks, the individual management and processing of license requests via a .dll interface, the combination of dongle or machine-based licenses, automatic time updates, or many other options: the system has the right answer for all of your requirements.

A single user interface under Windows makes for comfortable work with the different AxProtectors. The interface allows you to add significant protection against piracy, the theft of intellectual property, or tampering to your applications – requiring no special prior knowledge and only a few minutes of your time.

Security

This battery of protection mechanisms combined and the perfect coordination of all protection measures promise exceptional security, which can be improved on even further by means of individual expansions. Using this protection technology guarantees the correct usage of licenses and safeguards important competitive advantages.



Latest news summary

All New CmStick/M

CodeMeter's versatile capabilities, hard-ware-based AES encryption, and top flexibility are its hallmarks: the CmStick/MI is being released in 128 MB to 16 GB versions with a greater working temperature range and industry-class SLC flash memory chips. The CmStick/MC now offers high-quality 2-bit MLC flash from 8 GB to 128 GB. Prototypes are already available and the new models are now entering production. The new CmStick/M is ready for use even where mass-storage communication is prohibited. More details will be released in a dedicated whitepaper.

NEO Awards for Industry 4.0

The Karlsruhe Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB) is the winner of NEO2014, the Innovation Award from the Technology Region Karlsruhe. This year's NEO competition was focused on the concept of Industry 4.0, introduced by the German Federal Ministry of Education and Research, and was won by Fraunhofer IOSB's project "Secure Plug & Work." Wibu-Systems is a technological partner in this project and provides security for the automated configuration process in order for "Plug & Work" to become a reality for connected PLC components, sensors, and actors. Additionally, with its own Protection Suite, Wibu-Systems was itself one of the five finalists for the 2014 NEO Award. Picture: Fränkle

Blurry-Box® Cryptography

Wibu-Systems has teamed up with KIT, the Karlsruhe Institute of Technology, and Professor Jörn Müller-Quade, and applied the Kerckhoff principles to develop software protection processes that are reliably secure and correct even when their mode of working is known.

The Blurry-Box® design (patent pending) has won the first prize, including a 100,000 € endowment, for the biennial German IT Security Prize awarded by the Horst Görtz Institute in Bochum. The new concept will be integrated in new versions of CodeMeter.

Demonstration Partner at the Federal IT Summit

Wibu-Systems showcases CodeMeter as a demonstration partner at the "Industry 4.0 – Security Made in Germany" IT summit alongside such illustrious names as Infineon, Hirschmann, Telekom, Trumpf, and Fraunhofer SIT. The directors will signal the launch of secure manufacturing by handing over a CmStick to the German Chancellor Dr. Angela Merkel and the Federal Minister of the Economy Sigmar Gabriel. Industry 4.0 is the motto of the summit, which looks into improvements to productivity and the need for new security concepts, showcased here with a prototype made exclusively in Germany.

Great Reception of Webinars and Roadshows

Our regular German and English language webinars are becoming more and more popular. "Your driving license to expert cryptography" was one of the best attended ever. In addition to the globally available webinars, Wibu-Systems offers a chance for a personal meeting and intensive debates and discussions in our roadshows. Recent events in Europe, Wuhan, Xiamen, Hangzhou, Shanghai, and Beijing drew crowds of interested attendees.

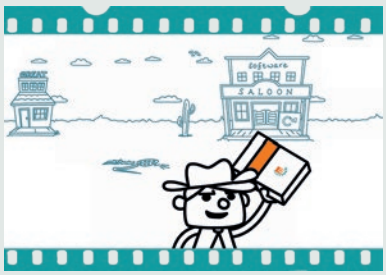
Using Certificates with CodeMeter in Windows 8.1

With charismathics' CSSI 5.0 PKI middleware you can save and use fully compliant Microsoft CSP or PKCS#11 certificates in CodeMeter in Windows (including Win 8.1), Linux, and Mac OS X. Dedicated solutions for OPC UA are being developed.

Whitepaper Virtualization

Readily scalable performance is only one of the many advantages of virtualization in business. At the same time, software developers still need to find secure licensing concepts for virtual environments. The new whitepaper explains the principles and the capabilities of CodeMeter and is available for download at www.wibu.com alongside a constantly updated list of interesting case studies.

Informative CodeMeter Introduction Videos



The second introduction video for CodeMeter has been released: **www.wibu.com/cml** While the first animation focused on the basic protection features, the second installment in the series looks at licenses, licensing models, and process integration. Spend less than two minutes to learn about CodeMeter's key features in a simple and entertaining fashion. Our verdict: a must-see.

Success Story custo med

The Challenge

custo med medical platform custo diagnostic consists of different software packages and devices in order to acquire the ECG, blood pressure profiles, and lung function data from patients. The products are integrated in clinical IT networks and need therefore to offer full modularity and scalability in terms of performance and usability. Furthermore, medical regulations require the after sales traceability and a protection against technological manipulation of the products.

The Solution

The various functionalities of custo med modular diagnostic platform are enabled by so-called protection bits or licenses in the source code of the embedded software. The transfer of the licenses is realized with customer specific license files, namely remote update files, created with a tool used by the sales department. The update file can only be applied to a specific CodeMeter dongle. Physicians and hospitals use a variety of internal or external CmDongles and CmCards, depending on the PC hardware. In large hospital IT networks, the trend is to use CmActLicense, the software-based protection solution.

The Success

CodeMeter allows for easy handling of complex and dynamic licensing models, even in demanding clinical environments, for a custom and straightforward license distribution over many locations. Furthermore, the licensing procedure can be efficiently managed when projects are realized in cooperation with authorized distributors. At any moment in time, custo med medical systems can be traced according to legal requirements. In case of technical problems, the specific release and license status of the user will be identified, allowing for a quick and safe after sales support.

The Customer

custo med is a leading brand in the medical diagnostic field. ECG, blood pressure, and spirometry data are acquired and integrated in hospital information systems. Perfectly tuned workflows for medical staff, dedicated software solutions, and highest standards concerning patient safety and comfort are the drivers of our product portfolio.

custo med was founded in 1982 and is located in Ottobrunn (Germany). Over 50 employees design, produce, and sell professional diagnostic software and hardware. The main markets are in Germany, Western Europe, Russia, Turkey, and Asia. All projects are realized in cooperation with qualified distributors.

Hans-Joerg Hoffmann,
Sales & Marketing Manager at custo med

The brand name custo med stands for a diagnostic cardio-respiratory acquisition and reporting system. A professional software design, combined with the convenient handling of the devices, provides smooth workflows in clinical wards. Thanks to CodeMeter, we have an unlimited range of action when it comes to offer individual license models – from single-user applications in medical practices to large hospital projects.

Software Licensing & Secure Code Seminars

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/trep**

Training location	Date	Time
Restaurant Het Wapen van Haarzuilens (NL)	2 Dec 2014	11.00-15.00

Contact your local sales representative for details		
United Kingdom / Ireland	+44 (0)2031474727	sales@wibu.co.uk
Netherlands	+31 (0)747501495	sales@wibu-systems.nl
Spain / Portugal	+34 (0)914148768	sales@wibu.es
Belgium / Luxembourg	+32 (0)34000314	sales@wibu.be
France	+33 (0)173030491	sales@wibu.fr

Meet with us:

Electronica 2014
Hall 5 | Stand 506 (at Infineon)
11.11.2014 – 14.11.2014
Fairgrounds Munich, Germany

MEDICA 2014

12.11.2014 - 15.11.2014
Fairgrounds, Düsseldorf, Germany

Bits&Chips Security 2014
Stand 1
19.11.2014
Brabanthallen Den Bosch
‚s-Hertogenbosch, The Netherlands

Bits&Chips Smart Systems 2014
Stand 11
20.11.2014
Brabanthallen Den Bosch
‚s-Hertogenbosch, The Netherlands

ESWC 2014
22.11.2014 - 23.11.2014
Hesperia Tower Hotel, Barcelona
Spain

sps/ipc/drives 2014
Hall 7 | Stand 660
25.11.2014 - 27.11.2014
Fairgrounds Nurmberg, Germany

Imprint
KEYnote
28th edition, Fall 2014

Publisher:
WIBU-SYSTEMS AG
Rueppurrer Strasse 52-54
76137 Karlsruhe
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
Rüdiger Kügler
Wolfgang Völker
Oliver Winzenried

Design
Markus Quintus

Print
E&B engelhardt und bauer,
Karlsruhe, Germany, EMAS III &
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 global-marketing@wibu.com

SECURITY
LICENSING
PERFECTION IN PROTECTION

WIBU SYSTEMS