FFI-rapport 2009/01137

Military operational systems in field – multiple levels of security

Tor Gjertsen and Nils Agne Nordbotten

Norwegian Defence Research Establishment (FFI)

23.06.2009

FFI-rapport 2009/01137

1086

P: ISBN 978-82-464-1602-1
E: ISBN 978-82-464-1603-8

Keywords

Sikkerhet

Virtualisering

Operativsystemer

MLS

MILS

Approved by

Anders Eggen	Project Manager
Vidar S. Andersen	Director

Summary

This report examines alternative solutions for handling information of different classifications on the same physical platform. The focus is on technology that is available today or that is expected to be available in the next couple of years. In order to support wide security spans, the solutions have to meet high assurance requirements and certifications. This is particularly the case if an Internet connection is to be included.

Both software and hardware solutions are considered. Trusted multi level secure systems belong to the first category as well as thin clients that can connect to different systems in a trusted manner. A hardware based alternative is to have a number of independent computers in the same case, eventually sharing a common console via a hardware switch. Another alternative is to have separate boot media, and reboot the system between the sessions. The last alternative in discussion is based on the MILS architecture (Multiple Independent Levels of Security). MILS is considered to support very flexible and future-oriented solutions, but have several limitations in the short term. It is discussed in the report how the main limitations can be circumvented in order to realize a solution. Related to an example coalition scenario the report outline some MILS based units for the tactical levels, from the coalition head quarter down to the individual soldier.

Sammendrag

Denne rapporten ser på alternativer for å kunne håndtere ulike graderinger på samme fysiske plattform basert på teknologi som er tilgjengelig i dag eller som er forventet å bli tilgjengelig de nærmeste par årene. For å kunne støtte et stort graderingsspenn er det nødvendig med systemer som er sertifisert til høyt tillitsnivå. Spesielt vil dette gjelde dersom en Internet tilkobling skal være inkludert.

De alternativene som vurderes er både programvare- og maskinvarebaserte løsninger. Til den første kategori hører betrodde, flernivå operativsystemer og tynne klienter som kan benyttes mot flere ulike systemer på en sikker måte. Et maskinvarebasert alternativ er å ha flere uavhengige datamaskiner i samme boks, som eventuelt deler samme konsoll via en fysisk svitsj. Et annet alternativ er å ha separate boot-medier, og starte systemet på nytt mellom sesjonene. Det siste alternativet som diskuteres, er basert på MILS-arkitekturen (Multiple Independent Levels of Security). Denne arkitekturen vurderes å gi de mest fleksible og fremtidsrettede løsningene, men har flere begrensninger på kort sikt. Rapporten viser hvordan en kan komme rundt de viktigste av disse begrensningene for å realisere en løsning. Rapporten tar også for seg et eksempel på et koalisjonsscenario, og gir skisser til MILS-baserte enheter for de ulike taktiske nivåene, fra hovedkvarter ned til den enkelte soldat.

Contents

1 Introduction

In military operations there is a need for handling information of different security classifications in the field. Traditionally this has been solved by using dedicated systems for each security level, and information has had to be moved manually across the air gap between the systems. However, this solution implies duplication in equipment and procedures, and is inefficient in operation.

Although far from ideal, this type of solution can be accepted in a tactical headquarter, but for highly mobile units in field it is more problematic. For vehicle mounted equipment, and in particular man pack equipment for the soldier, size, weight, and power consumption are critical factors, as well as the efficiency of the system to be able to react quickly in a critical situation. Depending on the role of the tactical unit there is a need to run different applications with different classification levels on one physical unit and to have mechanisms and tools to have a more flexible information flow between the levels, still limited by the security policy.

Because this has traditionally not been done, there are no established ways for doing this. This report therefore considers solutions that have the potential to enable multiple security classifications to be handled on a single mobile terminal. In some scenarios it is desirable that such a terminal also have a connection to the Internet, in which case an assurance level as high as EAL 6-7 according to Common Criteria will likely be required. If an Internet connection is not to be included, an assurance level of EAL 5 may suffice.

2 Alternative Solutions

In the following subsections we discuss alternative solutions and products that can potentially be used to handle information of different classifications on a single terminal.

2.1 Trusted operating systems

There are several operating systems with a strong emphasize on security. However, when looking at handling information of different classifications on the same computer, the assurance requirement depends on the security span. As mentioned, it must be expected that for highly classified systems also connected to Internet servers, an assurance level as high as EAL 6-7 will be required. In that regard, Solaris 10 with Trusted Extensions and Red Hat Enterprise Linux are only evaluated to EAL4 (both augmented with ALC_FLR.3 and conformant with the Controlled Access Protection Profile, the Role Based Access Control Protection Profile, and the Labelled Security Protection Profile). In fact, there is no general purpose operating system that is evaluated to EAL 6 or 7. The closest match is XTS-400 from BAE Systems, which is evaluated at EAL5. XTS-400 is further discussed in the next subsection.

2.1.1 XTS-400 (STOP)

XTS-400 is an MLS type of operating system and is a successor to SCOMP (Secure Communications Processor), XTS-200, and XTS-300. XTS-400 was first successfully evaluated at EAL5 (augmented with ALC_FLR.3 and ATE_IND.3) in 2005, and an updated version was again evaluated at the same level in 2008 (both times with support for the Labelled Security Protection Profile and the Controlled Access Protection Profile).

XTS-400 is actually a combination of the Secured Trusted Operating Program (STOP) version 6.4.U4 with specific hardware. The hardware included in the evaluation is based on the Intel IA-32 architecture,[1] and also includes several peripheral devices such as hard-disks, floppy drives, tape drives, video controllers, DVD-drives, keyboard/mouse, monitors, Ethernet cards, and printers.

The security target for XTS-400 [1] states that operating systems evaluated against that security target will:

- Associate sensitivity labels with all objects and all its users will have an associated clearance level identifying the maximum security level of data that they may access.
- Allow simultaneous use of the system by multiple users, all with different clearances and needs-to-know.
- Allow simultaneous network connectivity to networks of differing sensitivities/classifications (including IPv6 networks).
- Provide mandatory integrity protection of files.
- Provide an untrusted operating environment that includes common Linux commands and tools.
- Provide an Application Programming Interface/Application Binary Interface which is suitable for running most Linux applications in their binary format (no recompilation required).

To achieve this, the STOP kernel provides support for multitasking where each process is isolated in a virtual process environment. The security kernel running in ring 0 performs both mandatory access control as well as integrity control when a process is to access an object. The security kernel also provides I/O services and an inter-process communication message mechanism. Discretionary access control to the file system is enforced by the trusted system services running in ring 1, while the operating system services running in ring 2 provide a Linux interface to the applications running in ring 3. Although this provides a layered security model, there are also trusted applications, upon which the system relies to enforce the security policy, running in ring 3.

Because XTS-400 is targeted at server and guard applications, it is less suited for use on a mobile terminal. In particular, the hardware included in the evaluation is neither likely to be found in a standard laptop (or other mobile terminal) nor particularly suited for such use. A more recent

[1] CPUs: Intel Pentium III and Intel Xeon (P4) ”Prestonia”. Motherboards: Intel L440GX and Intel SW7501.

offering, STOP 7 [2], is to provide broader deployment options enabling tactical deployments. Although STOP 7 like its predecessor is to be certifiable at EAL5, no certification has yet been performed.

2.2 Thin Clients

There are several thin client solutions that can be used to access multiple computers within different security domains at the same time. Such products include:

- NYTOR's Trusted Multi-Net thin client [3] which enables accessing up to four domains simultaneously. It is based on a hardened version of Windows XP embedded and utilizes VMware to provide access to multiple domains.
- The Secure Inter-Network Architecture (SINA) thin client [4], from Secunet, which provides a thin client solution built on top of SINA Linux.
- The Sun Ray Virtual Display client which is part of Sun's Secure Network Access Platform (SNAP) [5].

The ultimate assurance levels of such thin client solutions are generally limited by their software implementation (including any underlying operating system). One way to mitigate this is by using a hardware solution for controlling the information flow to and from the thin client. Such hardware is provided by Tenix, with their Interactive Link Data Diode [6] and Interactive Link Keyboard Switch [7] (evaluated to EAL7 and EAL5 respectively). Using these, a thin client (or workstation) in a high classification domain can be connected to both a server in a lower classification domain and a server in its own domain. The keyboard and mouse of the thin client is connected to the servers in both domains using the keyboard-switch, enabling the user to select the active domain for the keyboard/mouse. Furthermore, the screen of the low classification server/workstation can be displayed on the high side thin client utilizing the one way channel provided by the data diode. This can for instance be combined with Sun's Secure Network Access Platform, as shown in Figure 2.1.

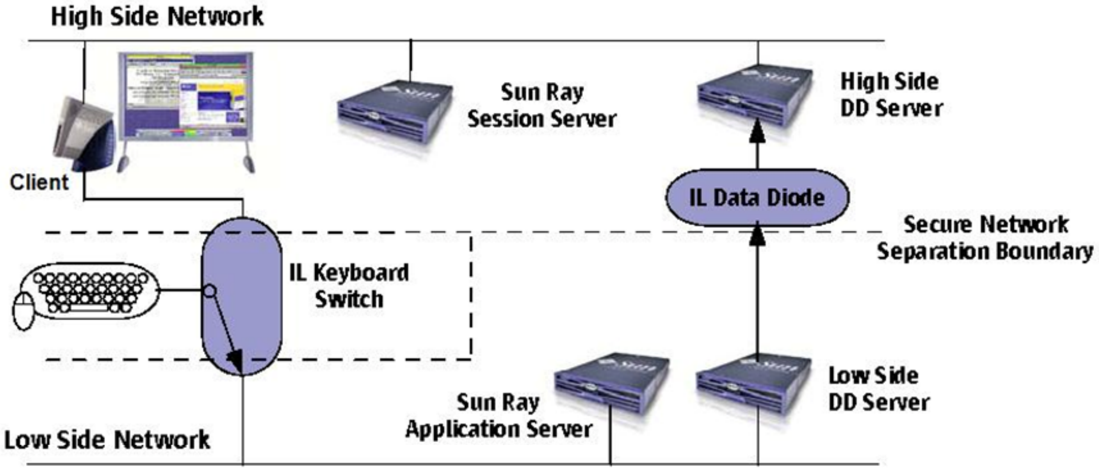


Figure 2.1 A Sun Ray Virtual Display Client (thin client) used in combination with the Tenix keyboard switch and data diode [8].

Nevertheless, because thin client solutions require a network connection to the server(s), they do not support offline use and hence do not provide a sufficient solution for mobile terminals with limited connectivity.

As part of the SINA product line there is also the SINA Virtual Workstation [9] though. Like the SINA thin client, the SINA Virtual Workstation is also based on SINA Linux. What differentiates the workstation from the thin client, however, is that it allows running multiple virtual machines each hosting a guest operating system (e.g., Windows or Linux), enabling offline use. Notice that such a virtual machine can also be used as a thin client by running the thin client software within the virtual machine. The SINA Virtual Workstation has been approved for use for multiple levels up to confidential in Germany and can be provided on both laptop and desktop platforms. However, the SINA Virtual Workstation does not appear as a candidate for higher classifications.

2.3 Multiple computers in one

A common way to handle information at different classification levels is to have separate networks and computers for each classification level. In this section we discuss a solution resembling this approach by having multiple computers in one.

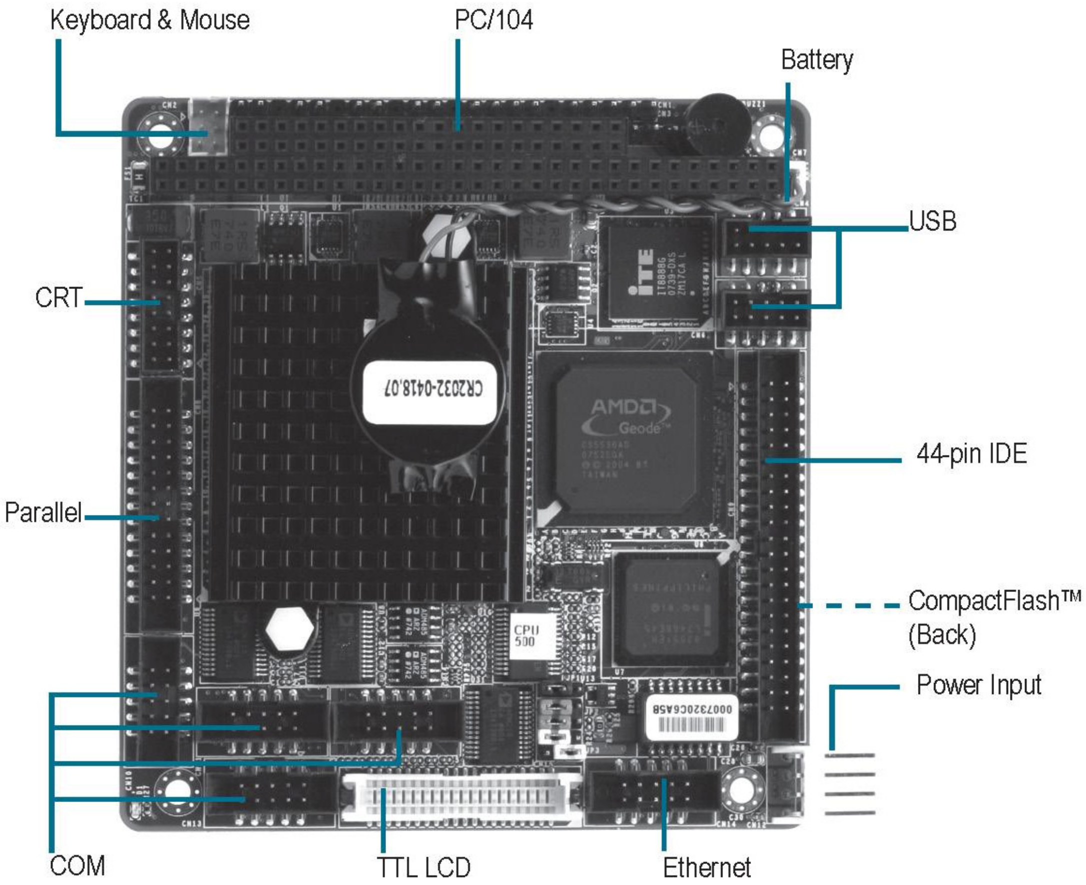


Figure 2.2 Single-board computer with physical dimensions 90x96mm [10].

A solution with multiple computers in one is facilitated by the availability of small single-board computers. For illustration, one such single-board computer (the PFM-540I from AEEON [10]) is shown in Figure 2.2. This single-board computer has an AMD Geode LX800, x86 compatible low power processor, capable of running Linux and Windows XP. The board also provides screen, mouse, keyboard, network, USB, and storage (IDE and CompactFlash) connectors, as well as support for up to 1GB RAM. The board does not contain a fan and is available in a version with an operating temperature of -40 to 80 degrees Celsius. Its physical dimensions are 90x96mm. More powerful single board computers suitable for demanding environments are also available (e.g., from GE Fanuc).

Due to the small physical dimensions, it is possible to fit multiple (e.g., two or three) such boards within a limited form-size, where each board is connected to its own storage (e.g., using flash disks/memory) while sharing the same power supply and battery in order to maintain small size and low weight.

A single keyboard, screen, and mouse can be shared through a built in KVM-switch. Alternatively, a more advanced solution could allow the screens of the lower classified domains to be sent through a one-way channel to the highest security domain, and be displayed within separate windows on that screen. This conceptually resembles the Tenix solution discussed in the previous section, but with the significant difference that an integrated one way channel would have to be provided. A third more rudimentary alternative would be to collect the keyboard, mouse, and screen cables from each board/computer within a colour-coded bundled cable/plug, where only one such bundled cable/plug can be connected to the screen, keyboard, and mouse at the same time. Finally, notice that because each board has its own network connector, each board can be simultaneously connected to separate networks/radios.

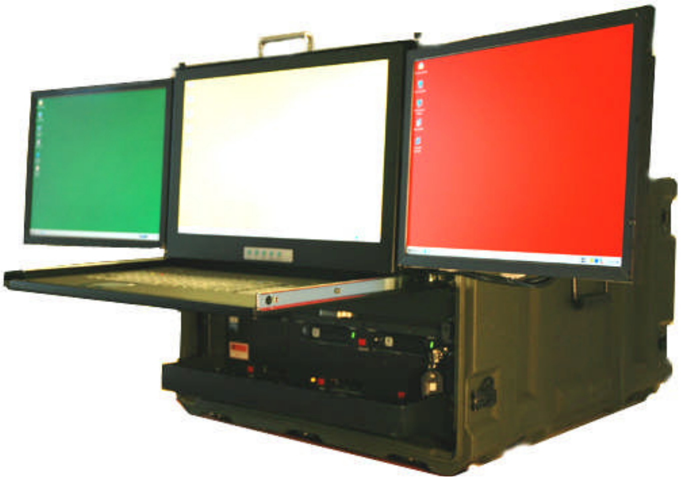


Figure 2.3 A DataVaultX4 tactical rugged mobile system [11].

When looking for existing products resembling the description given above, the closest match found was the DataVaultX4 from Secutor Systems [11] which integrates three (alternatively two) computers within a single case. Each hardware-based domain has its own screen, while a switch is utilized for sharing the same keyboard/mouse. Thus, each computer is separate and runs its own operating system (e.g., Windows). A version of the system was certified to EAL4 in 2005, and according to Secutor Systems there is an EAL5 evaluation pending. However, with multiple screens and a rugged construction, the form factor resembles more of a mobile workstation than a laptop, as can be seen in Figure 2.3. The system is also quite heavy, with a weight of about 40 kg. However, a laptop version is said to be under development [12] and there is also a version with a single screen.

2.4 Reboot based systems

If restricting the system to a more standard laptop or tablet-pc, the previous scheme is not applicable. In that case, one possible way to enable handling of information of different classifications is by requiring the system to be rebooted when switching security level. As there are many rugged laptops with removable hard drives, one option is to use a separate hard-drive for each security domain and reboot the system when a change of security domain is required. In order to save space it might also be an alternative to boot from memory sticks (without a hard disk present).

A more streamlined variant of this would be to have a laptop/tablet with multiple disks/memory-cards, where the user can select which disk/memory-card is to be physically connected to the motherboard using a switch. Such a switch could also be made to cut the power to the motherboard when changing disks, ensuring that the system is always rebooted when changing boot media.

Although an operating system supporting hibernation could reduce the time of these switches somewhat, such switches clearly represents significant overhead and would typically also require changing the network/radio connections and so on. Thus, such a solution only seems applicable when switches between security domains are relatively rare. With such a solution, it should also be assured that there is no covert channel from a higher to a lower session, e.g., via firmware memory.

2.5 Multiple Independent Levels of Security (MILS)

Multiple Independent Levels of Security (MILS) provides the means to have several strongly separated partitions on the same physical computer/device. As can be seen in Figure 2.4, the MILS separation kernel is fundamental to the MILS architecture. The separation kernel is basically a small piece of software that divides the system into separate partitions where the middleware and the applications are located. The middleware layer may provide an interface to the applications (e.g., a POSIX API and traditional OS services like device drivers and file system) or provide a virtual machine enabling an operating system (e.g., Windows or Linux) to be run within the partition.

The strong separation between partitions both prevents information from leaking from one partition to another, and also provides fault-containment by preventing a fault in one partition from affecting another partition. MILS also enables communication channels (unidirectional or bidirectional) to be selectively configured between partitions.

Although such functionality can also be provided to some extent using other software solutions (e.g., VMware or the SINA Virtual Workstation discussed in Section 2.2), those do not provide the same high-assurance.

The separation kernel concept has for a decade been used in real time operating systems (RTOS) where safety has been a strong requirement. The idea of using a similar technology for security critical systems was the basis for a program led by AFRL (Air Force Research Lab) in the US, which also introduced the term MILS. The goal of the MILS program was to support the evaluation and validation of MILS components, including the development of standard protection profiles (PP). The vision was to have a family of high assurance COTS products from different vendors that could be put together into complete systems of various kinds. The concept of compositional evaluation, where the evaluation of the complete system makes use of already evaluated parts, will then reduce the cost and time of the final evaluation.

For more details about MILS please refer to [13] and the references given there.

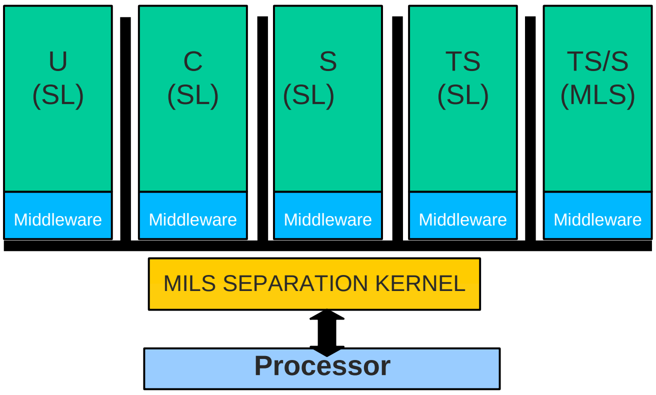


Figure 2.4 The MILS architecture.

2.6 Discussion of the alternatives

Of the discussed solutions, STOP and MILS appear as the most suitable software based alternatives for handling multiple classification levels on a single mobile terminal. By being more of a general purpose operating system, STOP in a sense provides the more complete solution. Although STOP enables running many unmodified Linux applications, it is nevertheless a non-standard platform that at least to some extent impedes application portability and reuse.

MILS on the other hand provides a more high-assurance platform, which will likely be a requirement for broader classification spans. By allowing standard operating systems to be run within partitions, MILS is well suited for application reuse. On the other hand, a functional MILS system may typically require additional functional components which are currently not available (at least not in an evaluated form). We will address this issue in more detail in the next section.

In light of the previous discussion it can be concluded that there is currently no complete high-assurance software solution for handling information of multiple classifications on a mobile terminal. However, MILS does provide a good platform upon which such a complete system can be built.

A hardware based approach based on multiple computers may provide an alternative to a software based solution. Such a hardware solution inherently provides a strong separation thereby facilitating system evaluation, but likely also incurs a higher weight and larger size (or alternatively reduced system performance). Having multiple physical computers in one also to some extent provides a less flexible solution compared to MILS. For instance, a hardware solution is less flexible with regard to the number of classification levels supported, does not facilitate for security critical components being protected within separate partitions, and does not provide a inherent mechanism for configuring the information flow between partitions/machines (although the latter could be achieved using special hardware as well).

3 MILS

We will in this section take a closer look on how MILS can be applied to military systems and discuss challenges to overcome. Clearly, a MILS workstation (as shown in Figure 3.1) will be a useful and attractive component that can solve some of the security challenges in military information systems.

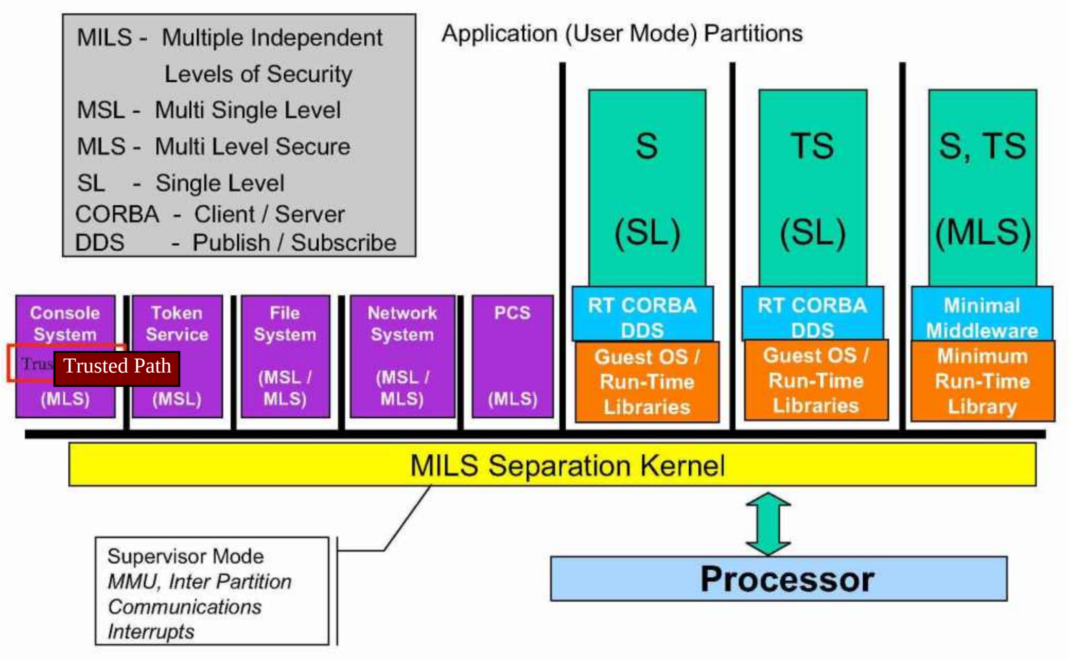
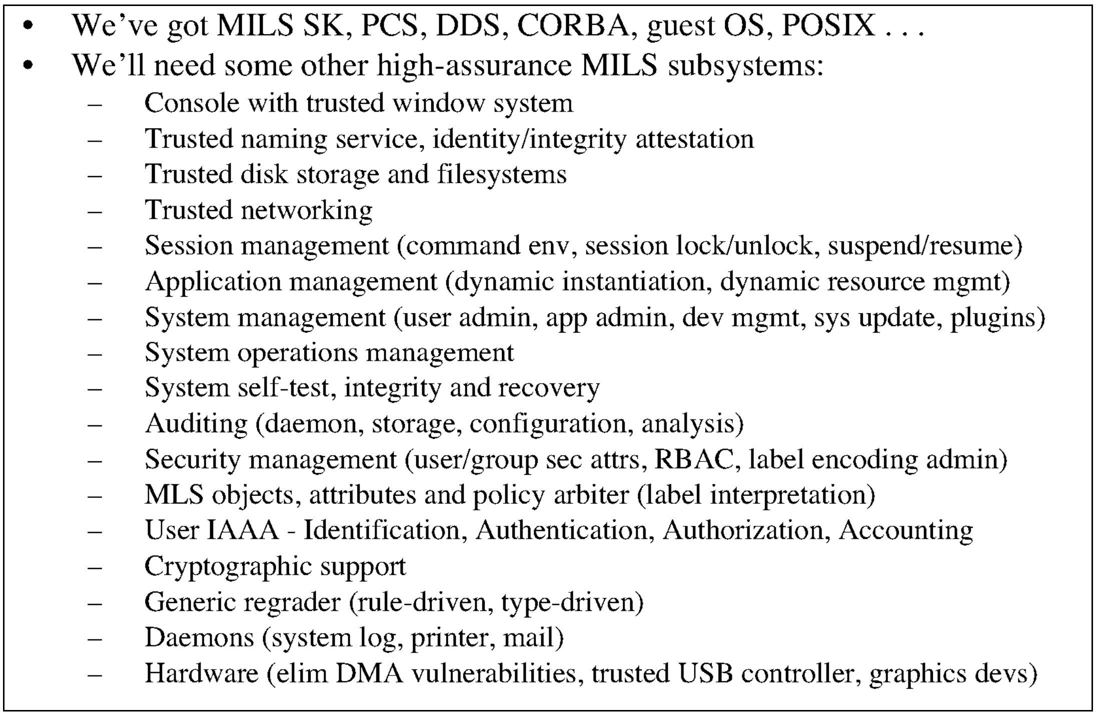


Figure 3.1 MILS workstation[14].

The complete MILS workstation consists of a number of MILS middleware modules for console, file system, network etc. which need to be evaluated separately, some of them at the same EAL-level as the separation kernel. How these are chosen will set the properties of the workstation. Such a streamlined MILS workstation is still a vision, however, since none of the trusted middleware modules are available yet.

An important prerequisite for the evaluation of these MILS modules are the protection profiles (PP). However none of them are approved yet (except for the Separation Kernel Protection Profile itself), but most of them exist as draft. Until these works are done and the evaluations are fulfilled, a full functioning high assurance MILS workstation is difficult to achieve. However, it is still possible to make use of the separation kernel to make a workstation with limited functionality. In the lack of standard certified MILS modules, some functions then has to be duplicated per level as single level modules, or they can be simplified to be able to do a preliminary evaluation.

Rance DeLong (LynuxWorks) lists what we need for a high-assurance MILS-based MLS workstation below:



- We've got MILS SK, PCS, DDS, CORBA, guest OS, POSIX . . .
- We'll need some other high-assurance MILS subsystems:
  - Console with trusted window system
  - Trusted naming service, identity/integrity attestation
  - Trusted disk storage and filesystems
  - Trusted networking
  - Session management (command env, session lock/unlock, suspend/resume)
  - Application management (dynamic instantiation, dynamic resource mgmt)
  - System management (user admin, app admin, dev mgmt, sys update, plugins)
  - System operations management
  - System self-test, integrity and recovery
  - Auditing (daemon, storage, configuration, analysis)
  - Security management (user/group sec attrs, RBAC, label encoding admin)
  - MLS objects, attributes and policy arbiter (label interpretation)
  - User IAAA - Identification, Authentication, Authorization, Accounting
  - Cryptographic support
  - Generic regrader (rule-driven, type-driven)
  - Daemons (system log, printer, mail)
  - Hardware (elim DMA vulnerabilities, trusted USB controller, graphics devs)

The list shows features (i.e., trusted modules) that are needed to fulfil the vision of a complete fully flexible MILS Workstation. However, it will take time to specify, implement and certify all of these. So, what can be done in the meantime? It is our view that it is possible to have a MILS-based system in operation even if the Separation Kernel is the only certified component so far. In the following the challenges to do so are discussed.

3.1 Technical Challenges

3.1.1 Certification of classified systems

For military systems to handle multiple security levels the technical implementation needs to be certified. Depending on the scenario and security levels there is a requirement for medium assurance (EAL4-5) or high assurance systems (EAL6-7). At these high levels a security evaluation takes into account both the software and the hardware of the system. This means that a certificate is valid only for the hardware that was the target for the evaluation. If a certified system is ported to another hardware platform, the system has to be re-certified.

A certificate stating an assurance level, EAL5 or higher is valid only in the country that issued the certificate. That means the actual product needs to be recertified by the security authority in another country. There is little experience in how to do this, whether the evaluation documentation will be released or not, and if so, accepted as is? That is a challenge, as a potential full re-evaluation will take time and be expensive.

3.1.2 Device handling

In a MILS system most of the traditional operating system functions are moved to the partitions. However, the device drivers need to be split in some way since the handling of device interrupts has to be done by the kernel, while the partition that is the “owner” of the device will do the rest of the device handling based on a “soft” interrupt from the separation kernel. The “owner” can be a dedicated I/O partition or a virtualized guest OS were the device is handled in a normal way for that OS. If more than one partition makes use of the device, the ”owner” has to coordinate the shared use and prevent conflicts.

The vision for MILS is to have standard and evaluated MILS components for functions that will share devices and resources, like console (screen/keyboard/mouse), file system and network. This work is still ongoing, and products will not be available in the near future. In the short term a solution to this problem is to have one device per security level (e.g. disk storage). However, if a truly mobile terminal is to be realized, it is not practical to have a separate console (i.e., screen, keyboard, and mouse) for each security level. The console issue is therefore further discussed below.

3.1.3 Console

A MILS based terminal/workstation for handling information of different classifications either requires a separate console (i.e., mouse, keyboard, and screen) for each classification level (i.e., partition) or a secure solution enabling the same console to be shared between multiple partitions. Although separate consoles may be an alternative in some cases, it does not provide a good solution for mobile terminals due to the resulting increase in size and weight. Thus, for truly mobile solutions, secure console sharing is required.

A basic console typically consists of three different devices, that is, an output device (screen), an input device (keyboard), and a pointer device (mouse). A more advanced console may have additional shared devices, but we limit our discussion here to the mentioned basic devices. Nevertheless, other devices can be handled in a similar manner. It should also be noticed that the discussion in this section only applies to devices that are to be shared between multiple partitions, as other devices can be handled through the MILS configuration.

In the following we sketch two alternative ways for console sharing that may be used as a starting point for a short term solution. The first alternative, shown in Figure 3.2, provides for displaying each partition (i.e., classification level) within a separate window on the screen. More specifically, each guest operating system (OS) used for handling information at some classification level is presented in a separate window on the screen. These windows may again contain additional windows in the case that the operating system (i.e., partition) represented by a window provides a windowing user interface. The screen partition itself may contain a stripped down operating system providing a window manager.

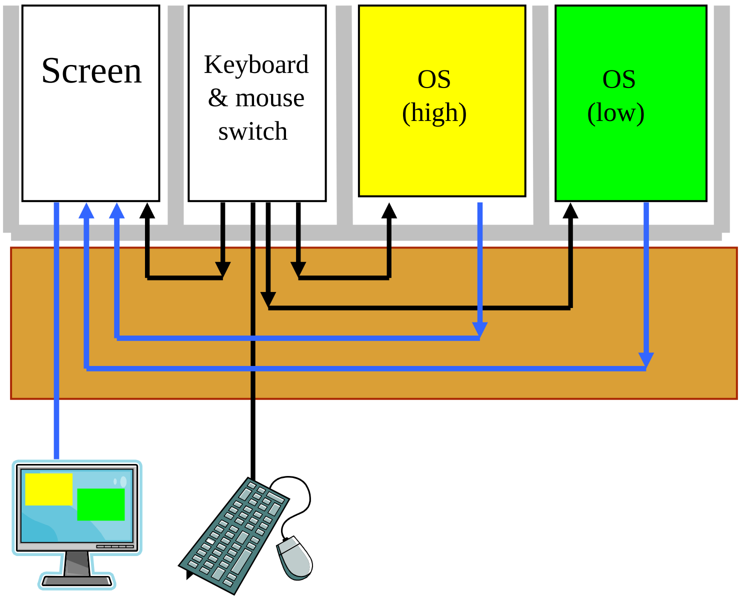


Figure 3.2 A console solution enabling multiple partitions to be visible on the same screen.

As illustrated in the figure, one way channels are provided from each of the OS partitions to the screen partition. These one way channels can be used for sending the display of each partition to the screen partition, e.g., using the Remote Framebuffer (RFB) protocol used in Virtual Network Computing (VNC) or the protocol of the X Window System. Because only a one way channel is provided, a proxy solution must be utilized. That is, each of the OS partitions must contain a proxy (e.g., implemented as a driver) that maintains any necessary state locally and provides the expected feedback to the local OS. This proxy would then transmit unacknowledged data to the screen partition, where the data is received by another proxy handling the communication with the window manager within the screen partition. As can be observed, this is somewhat similar to the Tenix based solution discussed in Section 2.2.

In particular, it should be noticed that because there are no outward flows from the screen partition to any of the other partitions, it is not possible for information to leak between the partitions through the screen partition. In order to reduce the risk of the user mistaking the windows corresponding to the different classifications, the windows should clearly show their corresponding classification level (e.g., using labels and colours assigned through static configuration). However, unless a high assurance operating system and window manager is used within the screen partition, there is still a risk that a window somehow could be erroneously labelled or that data somehow ends up within the wrong window. Considering that any information leak resulting from such mislabelling or misplacement would also require a human in the loop, this limited risk may still be acceptable for many usage scenarios in order to achieve a near term solution.

The other aspect which needs to be handled is the keyboard and mouse inputs, as these are to be shared between the different partitions (i.e., security domains) on a time basis. In particular, the keyboard and mouse are only to be connected to a single partition at any time. A key combination can be used in order to select the active partition of the keyboard and mouse. Hence, the software running within the keyboard/mouse switch partition needs to intercept the key combinations used to signal a switch to a different partition. When the keyboard/mouse is active within a partition, all inputs (except the key combinations used for controlling the switch) are forwarded directly to that partition allowing the user to interact within the window corresponding to that partition. In addition, the keyboard/mouse switch partition may also signal the screen partition when a switch is performed, in order to have the current active partition emphasized on the screen. By allowing the screen partition to also be selected as the destination for the keyboard/mouse inputs, the user is able to resize and move windows (i.e., the windows corresponding to partitions).

Because the keyboard/mouse switch partition is to be used with multiple partitions at different classification levels, the software for the keyboard/mouse partition is security critical. A high assurance implementation is facilitated however by the fact that this software only provides limited functionality. It may also be observed that the only input channels to this partition are from the keyboard/mouse, thus an incorrect keyboard/mouse switch software implementation would only be able to leak the keyboard/mouse inputs, and not information already within the partitions.

An alternative design is shown in Figure 3.3. Using this design only a single partition is shown on the screen at any one time. The console switch partition provides a single partition (i.e., OS high or OS low) with an indirect bidirectional channel to the screen partition at any one time. Because only one OS partition uses the screen at a time, there is no requirement to perform window management within the screen partition thereby enabling a simpler (i.e., more low level) implementation.

Because a bidirectional channel is established between the screen partition and the current OS partition, it must be ensured that no data remains within the screen partition when switching between OS partitions. One way to achieve this is to give the console switch partition permission to restart the screen partition. That way, the console switch can reset the screen partition when switching between partitions. At restart, the screen partition should be restarted from an immutable media and it should also be ensured that all its memory (including memory on the screen card) is cleared. By only having a limited amount of low level software within the screen partition, the time taken to restart the partition can be limited. (To avoid restarting the partition, a high assurance implementation of the screen partition software would have to ensure that no data remains between context switches.) As before, a keyboard combination is used to switch between partitions.

It is an alternative to connect the keyboard and mouse to the console partition instead of the console switch partition. The advantage with this is that it would reduce the complexity of the console switch partition, thereby simplifying its evaluation. However, a potential disadvantage with this is that it could provide a way to fool the user into believing that he/she is connected to another partition than what is actually the case. This could be done if the console partition and/or connected OS partition is compromised and somehow manages to either generate unintended partition switches or mask the users intended partition switches. As different partitions should have distinctively different looks (e.g., different desktop colours and so on), a sophisticated attack would be required to take advantage of this in a way that is not detected by the user. In particular, it should be noticed that the keyboard and mouse would still only be connected to a single partition at the same time, as this would be ensured by the (high assurance) console switch software.

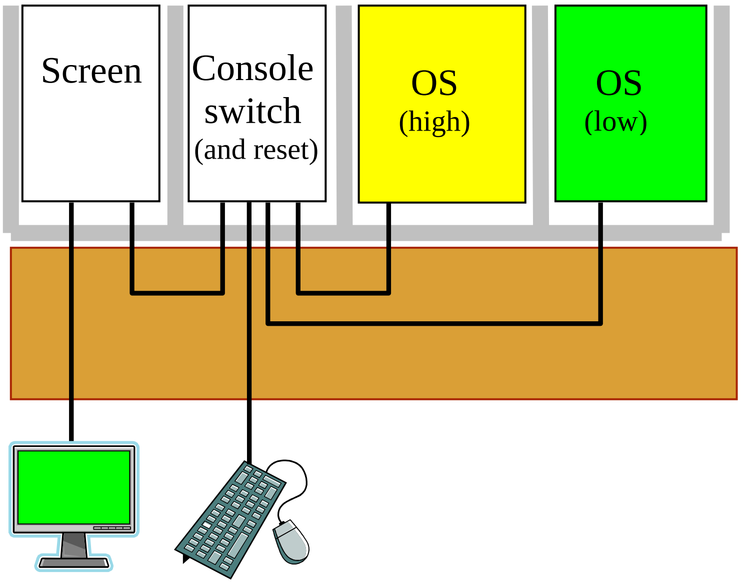


Figure 3.3 A console solution providing access to different partitions one at a time.

The motivation for the above designs is that the provided assurance should depend on the limited software within the switch partition (in addition to depending on the separation kernel itself), and not on the more complex screen partition. A disadvantage with the solutions above, however, is the additional processing overhead due to the additional partitions (including their contained software) and the communication between them. Considering that the mouse and keyboard generates very limited traffic in terms of bandwidth, the screen handling is expected to represent the largest overhead. This would particularly be the case if considering more graphic intensive applications, in which case a prototype would be advantageous to determine the actual performance impact.

For a desktop solution, the simplest solution in the short term may be to have a separate screen card and mouse/keyboard connector for each partition, and then use an external switch to select which partition is to be connected to the screen, keyboard, and mouse.

It may be observed that although we have discussed a solution where different partitions can be shown on the same screen, we have not addressed the issue of copy and paste between partitions. Such functionality is not inherently supported, but could however be added independent of the console solution. More specifically, copy and paste requires a channel to exist in the direction that data is to be allowed to be copied. Thus, when copying data it would be sent onto this channel by the initiating side. At the receiving side the copied data would then be collected from the receive buffer when a paste is performed. The channel for copying data could go directly from the originating partition to the receiving partition or through some intermediate partition for screening. This scheme can clearly be used independent of whether multiple partitions, or only one partition, are shown on the screen at a time. (If drag and drop were to be supported between different partitions, in the design where multiple partitions are shown on the screen at the same time, this would be much more complex.)

3.1.4 Storage

The concept for a MILS file system need to be flexible, since in a virtualized MILS system it has to support different guest operating systems. The solutions for this are outside the scope of this report. A certified MILS solution for use of the disk storage will perform partitioning of the disk to be shared by different virtual machines. In the lack of trusted MILS file system/disk partitioning mechanisms, a short term solution will be to have physical separated disks, one per security level/partition. It may also be observed that not all MILS partitions require a disk, as some MILS partitions (e.g., a console partition) may not require persistent storage.

3.1.5 Network stack

As long as there is a separate network interface card for each classification level, each partition can be connected to its corresponding network interface card thereby avoiding the requirement for a trusted stack. Still, there are plans for a trusted MILS network stack for proven assurance and robustness. Such a stack can be shared by multiple applications of different security levels to avoid duplication of network connections.

Because a trusted stack is not available yet, a MILS based system will need multiple network interface cards in the short term, one per security level. If they are to be served by the same physical network, the ports have to be encrypted individually, to maintain the strong separation of the levels.

3.1.6 Hardware platform

A MILS based high assurance system consists of the separation kernel installed on a physical hardware platform for which it has been evaluated and certified. A MILS system may also include standard middleware modules which have to be evaluated, some to the same assurance level as the separation kernel, but they will typically be hardware independent. Part of the evaluation of a separation kernel can be generic, towards some abstraction layer. The different vendors may have different strategies in this respect. Typically there will be some sort of a board adoption module to bind the separation kernel to a specific hardware, a processor mother board. A CC certificate will always be valid for a specific board or a family of boards, the latter to open for some variations of the hardware details. For changes in hardware a re-evaluation will always be required to get a valid certificate for the new hardware.

3.1.7 Available products in a short term

Integrity-178B from Green Hills has been certified for a motherboard with a PowerPC processor, as the first certified separation kernel. The second product that is in for evaluation is VxWorks MILS 2.0 from Wind River. This is also for a PowerPC type of board. The third MILS vendor, Lynux Works has not listed their LynxSecure yet (May 2009), but from what they have published on Internet it can be assumed that they target their planned evaluation to Intel VT type of processor. It is also expected that the Intel-based Integrity PC from Green Hills will be certified in some way within a couple of years. Integrity PC appears on Green Hills' home page as an existing secure product, but it is quite clear that a formal certificate will be required for security critical applications.

High assurance products imply the use of formal methods, also mathematical proof for the highest level, EAL7. Since the separation kernel is small, an evaluation is relatively cheap compared to that for a traditional monolithic OS, like UNIX (which in addition never will reach more than medium level of assurance, so the evaluations are not comparable). However an evaluation at the highest levels is still a big task and it takes time, at least 2-3 years to do an evaluation of a separation kernel. A re-evaluation of a certified kernel to another platform should be quicker if some of the material can be reused, and the vendor has been through the process before. In principle it is possible to pick a platform and have it re-evaluated and certified. In fact, vendors claim they have the expertise to do so, but they need customers that can pay for it.

3.2 What can be achieved in short term?

Short term solutions will be limited by what are available at the time of the implementation. For prototyping that is not necessary true, but for operational systems with high assurance requirements, all critical parts need to be certified. And for the next couple of years the only available MILS components will be the separation kernel targeted to PowerPC and Intel VT type of processor boards. Even with these limitations it will be possible to utilize the MILS architecture. In the lack of MILS middleware components, however there has to be single level partitions to support those functions. This will imply duplication of resources like one instance per level for disc storage and network stacks.

Also in short term some MLS modules may be required. Such modules can be implemented as "native" partitions, i.e., they are run on top of the minimum runtime, which is part of the certified kernel. Console handling (as discussed in Section 3.1.3) is one example, where an implementation of a trusted console switch can be the solution to avoid duplication of the console. Critical modules can be evaluated even in short term if they are small and with minimum functionality.

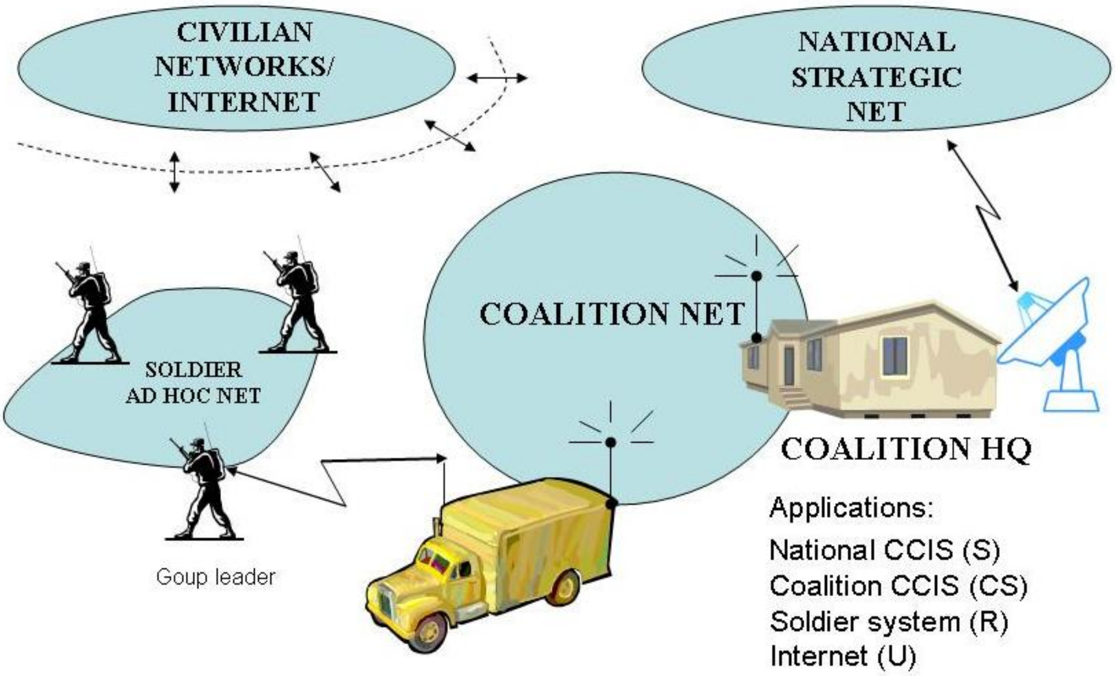
With an ambition level where medium assurance (i.e., EAL4-5 as opposed to EAL6-7 for high assurance) is sufficient, the certification process becomes easier. Partitioned real time operating systems are very similar to the MILS architecture, but with more functionality (e.g., device handling) in the OS. Such products include PikeOS from Sygso and LynxOS-SE from LynuxWorks, and similar products are also available from Wind River Systems and Green Hills Software. These products have been in the market for a decade, and support a lot of different hardware platforms/processor cards. They are evaluated towards safety criteria, and are claimed by the different vendors to be "easily certified" to EAL4-5. However none of these real time operating systems can be found in the lists of Common Criteria certified products. It is assumed they need a customer before they take that step. With a medium assurance level and a classification level of "Confidential" or above, an Internet connection can not be included on the same platform.

3.2.1 Example Configurations

In the following some possible short term applications of MILS in military systems are described, to support multiple security levels on one physical platform. To set the example configurations in a context, a scenario is drawn in Figure 3.4. This is meant as an example, and do not reflect a real scenario or requirement.

The configurations contain Internet connections. Existing NATO policy do not allow having connections to open Internet from a classified system handling "Secret" information. The reason might be that no traditional system has had strong enough security mechanisms, but they do not state what the requirements would be to have such connections. For the example scenario it is assumed that a separation kernel with high assurance certification will be sufficient for hosting Secret and Internet on the same physical platform, but for a real scenario this would have to be verified.

The figures indicate some of the flows set up in the separation kernel, but these are not complete. In particular, in the example configurations there are no flows drawn between the console partitions and the guest operating systems. That is because the console problem has alternative solutions in the short term, as discussed in section 3.1.3.



S=	Secret
R=	Restricted
U=	Unclassified
CS=	Coalition Secret

Figure 3.4 Tactical scenario – example.

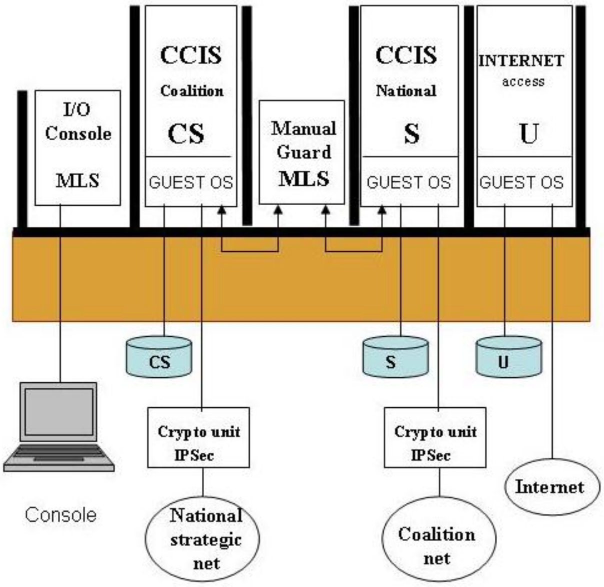


Figure 3.5 Multiple levels information system - Coalition Head Quarter.

The system for the Coalition Head Quarter is shown in Figure 3.5. It provides three different applications with different classifications, Coalition CCIS, National CCIS and public Internet. Internet is totally separated from the rest of the system, while information can be moved between the national and coalition CCIS via a manually controlled guard function. More sophisticated automatic filtering functions are possible to implement, but is out of the scope of this report. The Coalition Head Quarter has communications up to national strategic level as well as to subordinate units on the coalition network, all encrypted.

MILS based technology will show its full advantage when applied in highly mobile equipment, where small size and low power consumption are important characteristics. In the example configuration in Figure 3.6, the vehicle mounted type of equipment has connections up to the Coalition Head Quarter and down to the single soldier. The primary network for the soldiers is a short range ad-hoc net, but the leader of a soldier group is also part of the Coalition network, so there is some redundancy in the communication resources at the lower levels. The ad-hoc net in the example is a secure military 'WiFi' type of network. The coalition network is secured by separate IP encryption equipment.

The vehicle mounted equipment also has an Internet partition, in the case that Internet access can be achieved by means of some civilian mobile network.

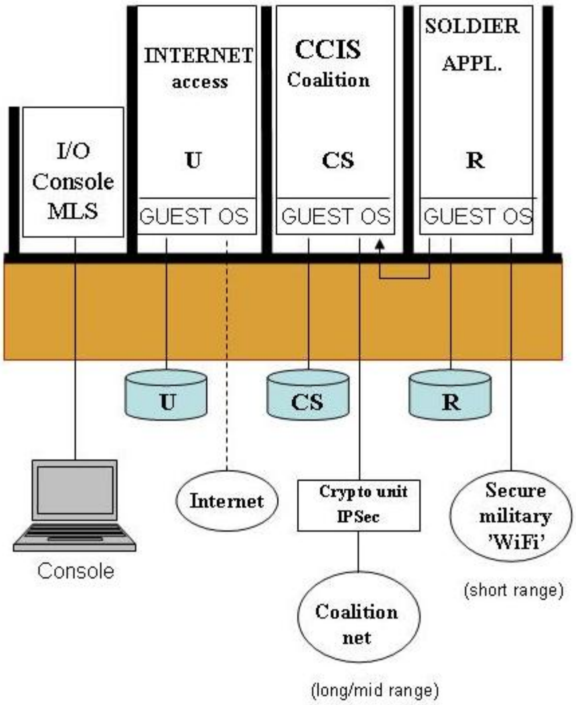


Figure 3.6 Multiple levels information system - vehicle mounted

The man pack type of soldier system for the group leader (Figure 3.7) has no Internet possibility. That reduces the assurance requirement, making it easier to find proper rugged low power hardware and have it certified. Medium assurance level is assumed to be sufficient. Apart from Internet, the functionality of the equipment for the group leader is similar to the vehicle mounted one.

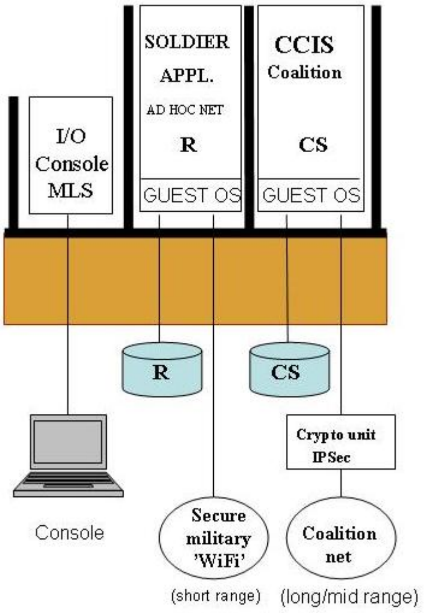


Figure 3.7 Soldier system - Group leader.

The equipment of the common soldier (Figure 3.8) is very small in size and provides the Soldier application solely. This will typically support simple mail, chat and voice services in addition to collecting a relevant situation picture for the group. It also contribute to the situation picture by delivering its own position to the other nodes. The configuration as shown is a single level system, and as such a MILS platform is not needed. However, if the system is connected to an unsecured network, then an encryption function has to be added, traditionally in the form of a separate encryption unit.

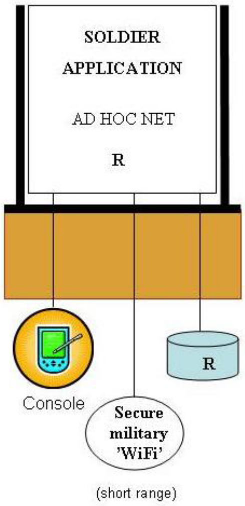


Figure 3.8 Simple soldier system.

A MILS based solution may facilitate the implementation of integrated encryption (Figure 3.9), which would be a very useful feature in mobile systems, to avoid carrying an additional encryption unit.

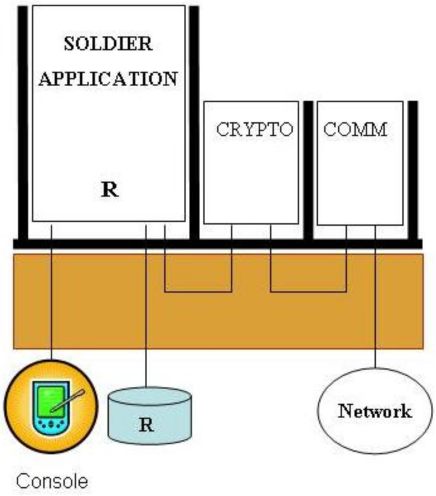


Figure 3.9 Soldier system with integrated encryption.

4 Conclusions

As discussed in the report, there are several potential solutions for handling information of different classifications on a single terminal. However, although there are both hardware and software based products providing this feature, existing products are not applicable for scenarios requiring truly mobile high assurance equipment.

A potential hardware based solution is to use several separate single-board computers integrated within a small form factor case. Unless such a solution becomes available from existing manufacturers, this will need to be custom built. A simpler design can be achieved by requiring the system to be restarted between session switches, at the cost of additional overhead in manual procedures and an increase in the time to perform session switches.

When it comes to software based solutions, the traditional solution has been to use an MLS operating system. If mobility is a strong concern, STOP 7 may currently provide the closest solution with flexible deployment options including man pack equipment. STOP 7 is however "only" targeting an evaluation level of EAL5, and such an evaluation has yet to be performed.

If high assurance is a requirement, then MILS seems to be the only applicable software based alternative. The MILS software architecture in principle resembles that of having separate computers, but with the benefit of providing a more space efficient and more flexible solution. Although the vision of MILS is to have a collection of high assurance off the shelf components that can be used in the assembly of various systems, only the separation kernel is currently available. Furthermore, as the evaluation of a high assurance separation kernel is targeted to specific hardware, only a limited number of processors/chipsets can be expected to be supported in the near future. Still, this report shows that the MILS separation kernel alone can be very useful when implementing high assurance systems. MILS based solutions are currently possible through duplicating devices, and through the addition of small custom certifiable components (such as discussed for the console). When additional types of certified MILS components are made available, they can be included in the system, to increase its functionality and avoid duplication.

References

[1] BAE Systems Information Technology, "Security Target, Version 1.22 for XTS-400, Version 6.4.U4," http://www.niap-ccevs.org/cc-scheme/st/st_vid10293-st.pdf.

[2] BAE Systems, "STOP Version 7, Platform for secure cross-domain application development," http://www.baesystems.com/BAEProd/groups/public/documents/bae_publication/bae_pdf_csit_xts_stop7.pdf.

[3] NYTOR Technologies, "NYTOR Solutions," http://www.nytor.com/solutions_listing.html#Products_anchor.

[4] Secunet Security Networks AG, "SINA Thin Client," http://www.secunet.com/fileadmin/Downloads/Englisch/Factsheets/secunet_SINA_Thin_Client_GB.pdf.

[5] Sun Microsystems, "Sun Secure Network Access Platform Solution," http://www.sun.com/solutions/documents/pdf/gv_snap.pdf.

[6] Tenix America, "Tenix Data Diode - Absolute Information Protection," http://www.tenixamerica.com/images/white_papers/datasheet_datadiode.pdf.

[7] Tenix America, "Tenix Interactive Links KBS," http://www.tenixamerica.com/images/white_papers/datasheet_kbs.pdf.

[8] Sun Microsystems, "Secure Network Access Platform Partner Ecosystem Architectural Guide," http://www.sun.com/solutions/documents/white-papers/SNAP_SolutionsGuide.pdf.

[9] Secunet Security Networks AG, "SINA Virtual Workstation," http://www.secunet.com/fileadmin/Downloads/Englisch/Factsheets/secunet_SINA_Virtual_Workstation_GB.pdf.

[10] AAEON, "PFM-540I Rev.B," ftp://data.aaeon.com.tw/DOWNLOAD/2007%20datasheets/ECD/PFM-540I%20Rev.B.pdf.

[11] Secutor Systems, "DataVault X4 2-in-1 & 3-in-1 Multi-Network MLS thin or fat client workstations," http://www.secutorsystems.com/WHITEPAPER2_6fSecutorSystemsApril2009.pdf.

[12] Secutor Systems, "FAQ," http://www.secutorsystems.com/faq.phtml.

[13] Tor Gjertsen and Nils Nordbotten, "Multiple Independent Levels of Security (MILS) - a high assurance architecture for handling information of different classification levels, FFI-rapport 2008/01999," 2008.

[14] Rance DeLong, "MLS with MILS?," http://cisr.nps.edu/downloads/invlect/delong.pdf.