



Reliable IT for production logistics

White Paper

7 December 2009

Copyright © 2009 by XR Systems

This document is for your information. Completely and unchanged print and storage is permitted. Further publication, re-print, duplication - in any way, in part or completely - can only be granted by written consent of XR Systems.

This document contains, apart from explanations, judgements and own investigations, descriptions of products, interfaces and concepts which are based on the publications of the particular manufacturers.

Reproductions of common names, brand names, trade marks etc. in this document are not subject to the acceptance that these names could be regarded as free or could be used by anyone, even without particular marking, in the sense of the trademark and brand protection legislation. All brands and product names are trademarks or registered trademarks of the particular mark holder.

In this document we have included links to the particular sites on the internet. For all these links we declare explicitly that we do not have any influence whatsoever on design and contents of these linked sites.

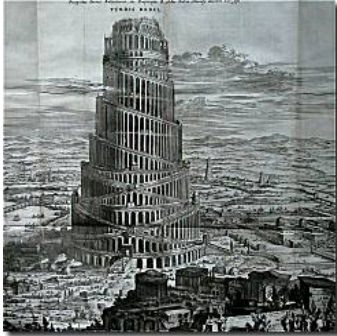
Hence, we dissociate ourselves completely and explicitly from all contents in all linked sites in our document and to not embrace any of their contents. This declaration applies for all displayed links in our document.



XR Systems GmbH
Münchener Str. 14
D-85540 Haar
fon +49 (89) 4272310
fax +49 (89) 42723170
net www.xrsystems.de

Reliable IT for production logistics

Preface



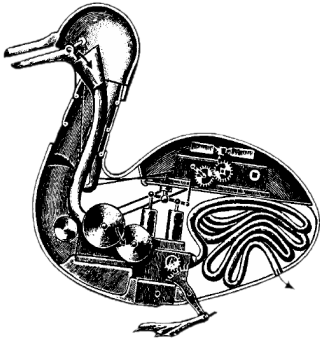
Each continuously successful IT project is based on a carefully tuned interacting of hardware and software as well as human resources for development and maintenance. Current hardware nowadays leads many grandmasters of the software trade astray and massive cathedrals of bone-crushing heaviness are created in a vacuum. In love with model and theory they loose track of the solution of the problem and render homage to checking [buzzwords](#) of high coolness factor.

It is the endless flexibility of software which nurtures the illusion that anything was possible - without paying any head to reliability and maintainability. Underestimating the permanently necessary knowhow for many projects in our time can be paving the way which leads into an intellectual deadend even before the monetary end which is bankruptcy. A long the lines of obscure products in the financial market, which had been triggering off the current economic crisis, dubious software strategies waste time and money on numerous projects. [Warren Buffet](#) used the term weapons of mass destruction in this context.

In the [production logistics](#) of well-known and internationally operating manufacturers DBtron/L has been one of the most [reliable](#) IT solutions for warehouse management and material flow control for over 20 years. Once in operation these systems work without interruption of over 1000 days and are only shut down for planned maintenance.

This [white paper](#) deals with the strategic basics of high quality and is intended for decision makers for and operators of extensive automated handling systems. It describes the technical system components and action principles which we have decided on. In the PDF version blue underlined links lead to additional information. In the download section of our website you can find all PDFs in different languages.

The facts described here are intended to be as simple as possible for an easy intuitive access to the subject. We would be glad about any feedback for improving this text.



Focus on Basics

The high quality in the operation of DBtron/L is due to the compliance with one single decisive rule which is like a mantra in all conceptual designs of our systems:

*Fight complexity
by
focussing on basics*

A well-known proverb is "Laziness is the driving force of all progress". It is however - do not get us wrong here - also the basis for the reliability of our systems. Of course, a lot of work and care is necessary to make a thing stable and error-free. But of what avail is it when we have to fight against the windmills of [complexity](#). Time and time again, it has been proven short-witted to try to deal with an inappropriate big number of matters at a time, most of all without proper reason and when dealing with highly complex IT systems.

Acting cool-headed here means knowing about the tempting complexity of the "want-all-now" and fighting it with complexity management. This does not mean to follow stupefying reductionism. Flexibility, the co-player of complexity, needs to be taken into account in appropriate ways. The flexibility needed in an IT project can vary depending on its later field of work. Thus a tailor-made customized solution for exactly one particular need will be far more limited than a standard solution which can be used in a big user market a thousand times more.

DBtron/L is made for the long-lasting world of production logistics and has already been used by several customers for over 2 decades. Consequently, different approaches and decisions are required as in comparison to the system for the one-time reorganisation of the German post codes after the country's unification.

System properties considered in long-term are amongst others:

- functional complexity of the system
- full development
- stability
- flexibility
- procurement costs (Make or Buy)
- usage costs incl. support

In the following we will explain which key components are used in DBtron/L and why we think these are the best choice.

Technical System Components

Database

From the very beginning for DBtron/L we have integrated only one database which is technologically leading: [Oracle](#).

It is the state of technology to base a comprehensive business process software on a powerful [database management system](#) (short: database¹). Again and again software developers have to make the same elementary decision i.e. which (one) of the current systems in their software architecture they ought to support.

In 40 years of their development databases have grown into powerful and bulky entities due to permanently increasing requirements. Despite efforts to standardize the systems the differences between them are bigger than the features they have in common. Technological progress has long overtaken any standardisation respectively has limited all to a small nucleus of database functions.

In the mid eighties a deciding factor was that our small but highly motivated team of software developers was given the chance to support a globally operating company in the derived-timber industries with a basic release of DBtron/L. So we could bring in, then as now, our extensive expertise in the leading database Oracle.

Databases during the 1980ties

The market of databases was completely different compared to today's:

There were numerous competitors; however, high prices made painful investments on customer side necessary and only highly skilled IT specialists were able to manage those systems. Quite often companies decided strategically for one company-wide database.

Thus it seemed logical to make software applications compatible to the current and already existing databases of the users. In order to maximize the sales of a software company it could seem logical and vital to develop as neutral as possible in relation to many of the sometimes expensive systems.

1) The term database, database server, database system and database management system can be used equally without significant differences in the IT.

The Illusion of Independence

To support more than one data base increases expenses far beyond the savings hoped for.

If the software developer goes for a clearly arranged and easy to maintain code base of the software application, he/she needs to do without extensive database functions that are beyond a small standardized nucleus. Consequently, he/she has to simulate a big part of modern database functions in his/her software application. At the end, a considerable amount of work on the part of the developer is necessary and/or components of third parties are to be integrated. Both versions will effect the complete [Total Cost of Ownership](#).

Database functions relevant for DBtron/L beyond standard are amongst others:

- regulations of competing data access in multi-user operation
- restart and data recovery after system failure
- aggregation of data for management reports
- archiving of data and their change log for back tracking
- report and back tracking of system activities
- system performance (speed, capacity)

New dependencies, which are to be avoided, accrue from any abandonment of database functions; however, these dependencies should never take place for more expensive or less durable resources² than that of the database system of a world market leader.

Databases today

Two decades after our first DBtron/L project the market of data bases has changed completely since.

Undergoing a period of consolidation there is just a small number of relevant³ competitors left of the many that existed years ago. Oracle is the undisputed leader with a 46% market share and stronger than the next 3 competitors altogether.

Nevertheless, depending on the intended use⁴ prices for licences of these systems have decreased drastically and you hardly notice them in the overall costs.

2) incl. software developers

3) in the long run suitable for fundamental enterprise applications

4) e.g. as embedded (encapsulated) license within application software

The Oracle Database - an inherent part of DBtron/L

The focus on the data base Oracle enables us to make use of the entire functions of the powerful data base software and to maintain an encyclopedic knowledge in the administrative usage. Due to developing tools that are either directly of Oracle or are at least optimized for the Oracle data base we can reach best possible productivity. Consequently, in the long run, the user receives a solution which satisfies his/her needs most reliably and most cost effectively.

In an important enterprise business software the database is *the* most crucial component of the system. Here the most valuable business data is entrusted in. Due to this reason the decision for a database in long-term use will always be the most permanent. During two decades cycles of innovation entailed several generations of computer systems and system software, methods of software development and programming languages for the DBtron/L user and developer; just like a remote island the Oracle database survived all changes and saved investments and data of its users.

Now as well as in the future Oracle will be available for new and marketable systems (see Linux) and will guarantee the protection of customer investments.

Hardware - better virtual

Virtual hardware saves the most valuable resources of an enterprise: your members of staff - thus saving expenses and increasing the system quality.

In the hardware- and software-scape a striking change took place, too. Twenty years ago the ORACLE database was available for over a 100 different computing platforms (hardware, operating systems). By now the diversity of proprietary systems, that used to be, gave way to a strong standardized landscape of computers and operating systems⁵.

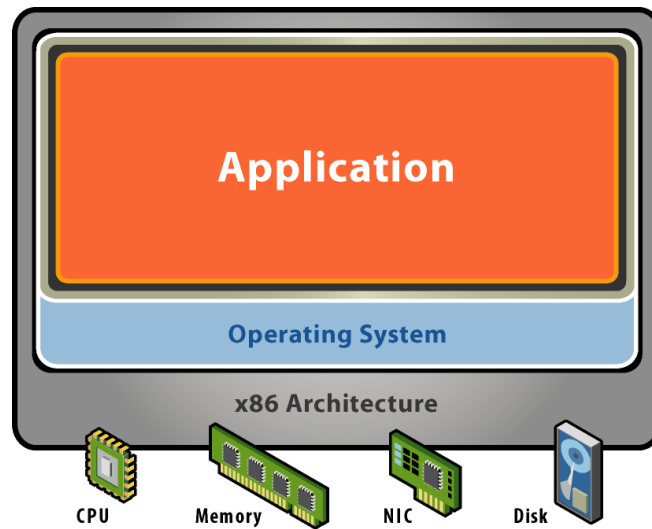
Conventional hardware

However, it is still rather complex to make an application software available for different operating systems. Consequently many programs are either fully functional for only one particular operating system or are at least optimized therefore.

The bottom line is that there exists an uncontrolled growth of data processing servers, sometimes one per application even. On top, different operating systems multiply expenses for qualified operators.

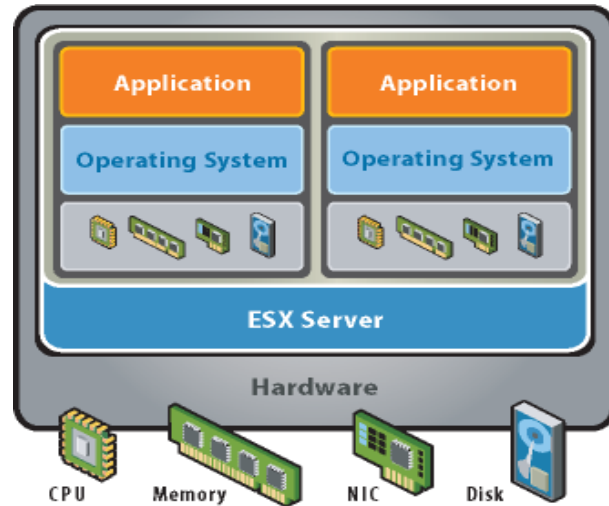
5) measured by sold systems, systems with Intel and AMD (x86, x64) processors are leading the market.

When handing over an IT Project the tricky question of the customer (user, operator) emerges regularly for an appropriate and permitted operating system.



The close link between application software, operating system and the hardware below made an installation and configuration laborious until now. When servers were down and/or servers were renewed these jobs accumulated causing costs and standstills.

The virtual computer



The long-lasting process of IT standardisation currently culminates in the virtualization of hardware. [Virtuality](#) means the property of a matter not existing in its obvious form but to be similar to the latter in its entity or its effect. Virtual computers - also called virtual machines (VM) - are not made of hardware but are simulated by software.

Virtual machines are controlled and are as far as possible isolated operation islands for pre-installed, pre-configured and at-once usable software applications; they are already configured with an operating system (guest), have been packed in an executable format and are runnable on a prepared server (host) with an appropriate virtualization software.

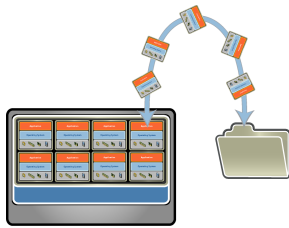
In this context the term [Virtual Appliance](#) has been accepted which refers to the comfortable use of home appliances: plug in and start.

Flexibility

Virtualization of hardware makes independent from the knowledge of operating systems of the employees.

Sometimes another operating system can exist inside of a VM than outside or what it is based on⁶. The software for virtualization creates standardized conditions (virtual processor, virtual storage devices, virtual network interface cards etc.) inside the VM and thus making copies and shifts of the VM between servers possible.

6) normally hardware and software architecture is described from bottom (physical hardware) to top (user interface)



A VM has been packed in a few⁷, however, some very big data files. But due to the immense growth in capacities of mobile data media and in network bandwidth an efficient transport of sometimes hundreds of gigabytes is possible.

A perfect clone is created by copying the files of a VM. As you are independent of a physical server you have decisive advantages for provisioning, service and a safe operation of software applications.

Perfect Quality of Delivery

By creating an absolutely identical copy of a VM you get around an intensive and error-prone installation job just by a trivial technical operation.

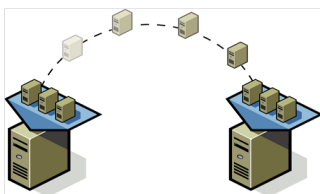
As customer you will get an exact clone of precisely this VM, which has been continuously improved and quality assured in our development department over a long period of time. All findings gained during operation and development are integrated in the master VM and are passed on completely to each of its clones.

The preserving value of intellectual work will be maximized and automatized: errors in configuration are avoided. Accurate notes of changes and even most meticulous compliance cannot produce any comparable quality.

Quick move during server breakdown

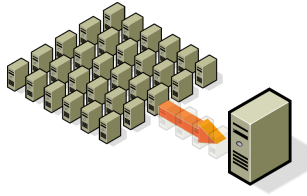
Computer hardware can and will break down - wear-out and ageing are definitely the major weaknesses of all materials. Hardware virtualization avoids and/or shortens interruptions of operations.

Modern virtualization technology is an essential part of our availability strategy. As we stated above, a virtualization of servers levels the differences between guest and host as well as between different host systems.



Thus we have been able to move VMs between servers which have had several years of difference in development. Even a notebook equipped accordingly can sometimes be home for one or more VMs for the time being. Depending on the importance of the software application and depending on the budget a move of the VM can even be managed while running.

7) If applicable just only one



Consolidation

The maintenance costs of a server during its life-span (cost-of-ownership) exceeds regularly its initial value. Virtualization cuts down on the hardware outfit in the computer centre.

The capacity of a server is usually sufficient for more than one VM. Software applications can now be operated on one mutual hardware⁸. Before, software applications had to be on different servers only due to their different needs according to their operating area (operation system, versions, etc.). Often these servers were not even running at full capacity then.

VMware - virtualization for enterprise class applications

When virtualization is as important as it is for us in DBtron/L, the chosen system software has to meet crucial criteria.

[VMware](#) has been the pioneer which, more than 10 years ago for the first time, brought the virtualization technology from costly main-frame systems to well-priced commodity hardware. As the market- and technology leader VMware has the widest range of products and is the best choice for enterprise critical applications.

Since 1999 all DBtron/L installations have been based on VMware and since 2005 on Version ESX which is suitable for data processing centres.

Operation systems Unix and Linux

Unix/Linux is still leading when it comes to reliability - Linux ontop offers the best price-performance-ratio

[Unix](#) and its many derivatives are regarded as most effective and reliable operating systems by professionals. They are still *the* preferred method of choice for running essential corporate IT applications when appropriate hardware is available. All earlier DBtron/L installations were based thereupon⁹ and some are still running today.



[Linux](#), which is a free and open source copycat of Unix, counts for many as cost-effective alternative for Unix as well as [Microsoft Windows Server](#) and has made it into sensitive corporate areas. Since 1999 Oracle has supported Linux with up-to-date versions of the database and even develops on this platform now.

8) Aspects of isolation and mutual interference would go beyond of this paper. Please contact us for further details.

9) Hewlett-Packard HP-9000 with HP-UX

For this reason it was a logical and evolutionary step for us to go from Unix to Linux. For over 10 years all deployments of DBtron/L have been based on the distribution [Red Hat Enterprise Linux](#) respectively on its derivative [CentOS](#).

PLC activation without OPC

Reliable material flow management systems are dependent on the stability of the PLC interface.

When linking up to PLC systems, which are the control level, we are getting into a dense area. Instead of supporting numerous¹⁰ incompatible products we concentrate on the market leaders Siemens (S5/S7), Allen-Bradley (PLC5/ControlLogix) and AEG Modicon.

DBtron/L communicates directly with the PLC level without any further complex intermediate level which itself has to make use of more complex failure-prone components.

In comparison to competitors who use difficult [OPC](#)-servers as link, XR Systems uses in Dbtron/L a comparatively tiny software library and exchanges data with the PLC in the shortest and most stable way.

10) Market observers know over 300 brands worldwide

Action principles

Business logic in the data base

Over 20 years of experience in using Oracle have shown us time and time again: the database is the best place for implementing a function.

Comprehensive area of programming

Not only is the safe and flexible administration of any kind of data one fundamental highlight of Oracle but also is the storage of the program code¹¹ in the database and the execution of them in close range.

Here Oracle's potentials go far beyond those of competitors and can be used as comprehensive environment for the implementation of your business logics, completely.

We have generated most of our program code within the database in DBtron/L, too. With any migration of the database to another computer platform the execution environment moves along with the program code - even when processors or operating systems change.

Traceability

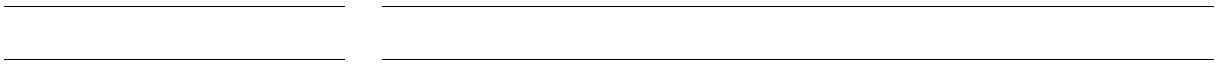
In order to be able to trace challenges at a distant place in the world on the customer's site we have always kept a comparable configuration in our development department. This used to include complete computer systems along with identical versions of the operating system and the database.

This has become much easier and most importantly cost-effective due to hardware virtualization.

Responsibility

We are aware of the importance of our IT systems for every single user. Each of our system developers shows a maximum of identification with the productive operation and takes this responsibility in support cases personally. An unnecessary delay of a solution of the

11) Written in [PL/SQL](#) or [Java](#)



customer's problem can thus be avoided as there is no gap between supporter and developer.

Summary

The [Pareto principle](#), also called Pareto effect or 80/20 rule, means that 80% of results are achieved in 20% of the overall time. The remaining 20% cause most of the workload - and most of the errors.

By consistently focussing on the essentials we are striving for a perfect solution for the IT systems used by professionals in the industries and this by balancing all costs (also in the think tank) and the usefulness (flexibility, reliability).