ArchivistaBox 2014/IX with all-new ERP solution
Egg, 28. August 2014: *The ArchivistaBox has included ArchivistaERP for quite a while now. Customers who use ArchivistaDMS can activate the module with just two clicks. To date, however, ArchivistaERP has been running in the background. The ArchivistaBox 2014/IX changes this: ArchivistaERP is now equal to ArchivistaVM and ArchivistaDMS. The following blog introduces ArchivistaERP and gives an overview of its development history.*



What happened so far…
If you read our blogs from **time** to **time,** you may have come across **one or two articles on ArchivistaERP.** ArchivistaERP was created in 2008 as an alternative to the three different invoicing systems that our company had come to use over the years (purchased solutions and in-house developments), which did not communicate well with each other. We **wanted to replace those solutions with a single open-source, web-based ERP solution.** For a software company such as Archivista GmbH, this shouldn't be all that difficult. With this guiding principle in mind, our company invested many working months in evaluating the previous solutions, **SQL Ledger** and **TinyERP (OpenERP and Tryton today).** Unfortunately, TinyERP was problematic: even though this RichClient application did have an appealing look, it contained several defects (particularly for printing).
SQL Ledger ran fairly smoothly but was not adapted to conditions in Switzerland at the time. In a nutshell, the attempt to adjust these solutions to our needs ended in disaster. We ended up with a six-figure line count, frames, a lack of security management — in short, maybe it was precisely because we attempted to adjust the code that we abandoned the project in the end. Through this (somewhat painful) learning experience, we realised that open-source solutions, too, only have value when they are developed accurately.
Representing many voices, there is an article on **lwn.net/Articles/227151/** containing the following quote: **"…SQL-Ledger's code is absolutely horrible".** Another quote: **"…not only the code – the UI is imho unusable."** The SQL Ledger experience further taught us to handle our own code in a more discerning manner. Many things may have gone into production with a slight delay since then, but there was never a risk of a deterioration of quality on this code base. We cannot make a conclusive statement regarding the situation with non-open-source ERP systems. But I am no longer willing to evaluate any solutions whose source codes are being kept a secret by the

manufacturer.
About one year after terminating our attempt, we started a new approach with **OpenBravo** and **FrontAccounting.** We would have approved of OpenBravo, but in the end it seemed far too powerful for an SME. **FrontAccounting, on the other hand, proved more lightweight and we were able to implement the resulting ArchivistaERP at our company in 2009.** The code was significantly more useful than that of the SQL Ledger (even though we first had to learn PHP). Only the fragmentation of the code (approx. 360 files) proved a difficult obstacle – such a difficult obstacle, in fact, that any further development was repeatedly postponed. This summer, we were finally able to update ArchivistaERP, and the result can be found in the ArchivistaBox 2014/IX.

Completely new code base

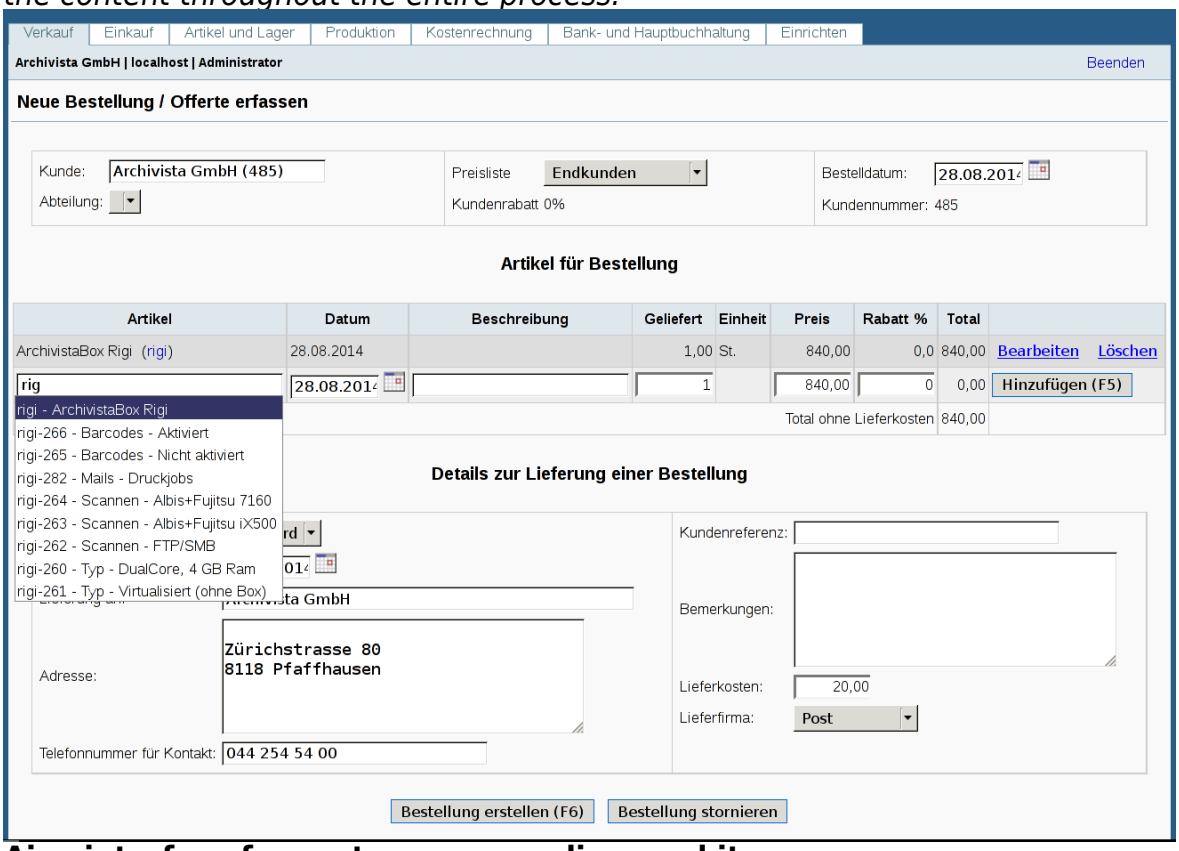
Firstly, the source code was completely overhauled. The **approx. 360 files were migrated to one file and approx. 60,000 lines of code turned into 28,000. All forms and reports were categorised into classes** and can now be reused as needed for customer projects. This means that a base library and approx. 100 forms and 30 reports are currently available. These classes can be expanded for projects as needed. All (class-based) forms and reports can be called up via 'index.php' (e.g. index.php?form=order). Another new feature is access via HTTP requests or directly via PHP (in batch mode). Here is an example:

```
 $_POST['batch']=1;
$_POST['login_user']="admin";
$_POST['login_db']="archivistaerp3";
$_POST['password']="archivista";
include("session.inc");
include("form_order.inc");
$customer=345;
for($c=1;$c<=10000;$c++) { // replace $c<=10000 with $c<=1
for one order
  $form=new form_order("NewOrder=1");
  $form->form_order("customer_id=$customer");
  if (getCartVal('customer_id')==$customer) {
$form->form_order("stock_id=dolder&price=310&qty=2&AddItem=1
");
$form->form_order("stock_id=eiger&price=4020&qty=2&AddItem=1
");
    $form->form_order("ProcessOrder=1");
  }
  $form=null;
}
?>
```

This code is used to **store orders on the system in batch mode. The process executes the same code as is run when a logged-in user records an order via the web interface.** The order can also be transferred by HTTP request so that other programming languages are not excluded. What benefits does a batch mode have for me as a customer? Well, as a customer, I receive a simple interface to the application (API) without having to understand the internal structures of the application. So the date field, for instance, is filled in automatically during the order, and I can also access the internal price structure (end customers, resale, etc.).

Just a note on the side: *it could be said that one of the greatest "sins" of the PHP language is that the screen display is triggered more or less from the first character, due to the HTML embedding. This can make HTML artefacts harder to read, or result in large numbers of echo commands in the PHP code. Specifically regarding ArchivistaERP. To date, there were over 3,600 echo commands; now there is only one call-up. This allows the application to decide how to deal with*

the content throughout the entire process.



Ajax interface for customers, suppliers and items

It used to be an annoying feature of older web applications that information was transferred to the user in a value1;value2;... format. Unfortunately, ArchivistaERP is no exception. This is still manageable where the scope of the customer master data does not exceed a four or five-digit address count, but with several million addresses, this would be an insurmountable problem, as all entries would be sent to the web server for each page call-up. **The integrated Ajax technology only transmits the customers, suppliers and/or items that are currently being processed, without having to reload the pages again each time.** Just enter the first letters and you can select the right customer, supplier or the requested item.

Function keys for fast working

In many cases, data can now be recorded in ArchivistaERP by means of the function keys. To create a new order, for example, you can use the F2 key, then select the customer directly, use the tab key to get to the items, where you can use F5 to move items into the current order and, finally, you can save the order with F6. This way, an order can be created in just a few seconds.

Does an ERP solution really need 32 or 64 GByte of RAM?

When global corporations rely on PHP, PHP really can't be slow. But let's be honest – run-time speed is rarely the problem with today's programming languages and computers. Nonetheless, many applications are still slow or merely performant when extremely powerful computers are used. An ArchivistaVM customer ordered servers with 128 GByte of RAM because the new ERP solution (a product by that company with the tiles) required 64 rather than 32 GBytes of RAM. FYI: this customer's daily ordering procedure amounts to several hundred invoices.

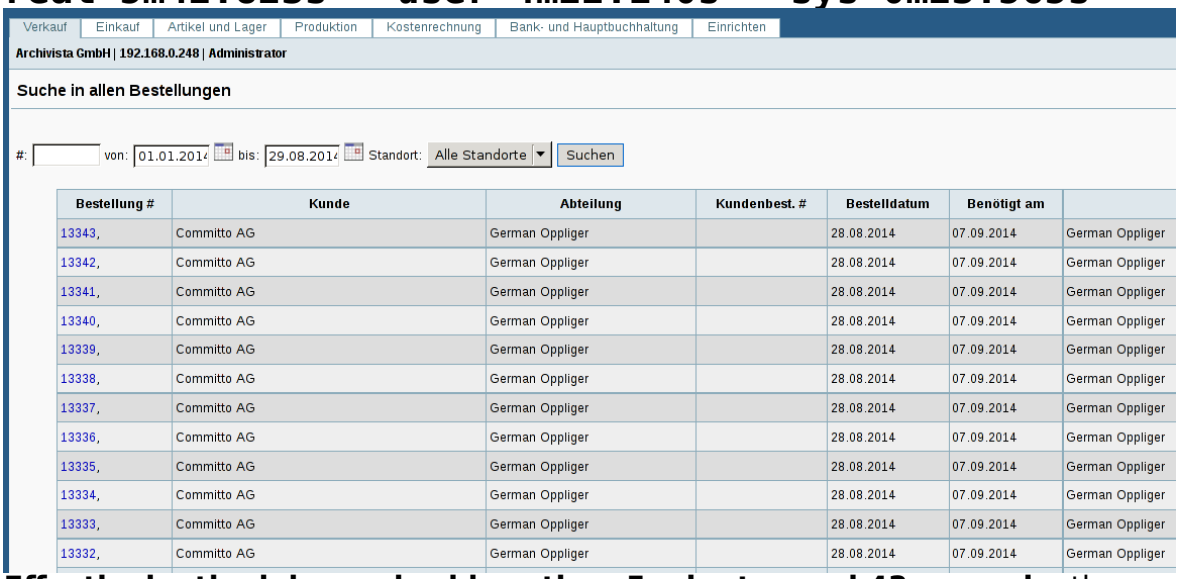
With ArchivistaERP, we **placed great importance on making the application faster than fast.** To test this regularly, we used the **ArchivistaBox Albis III** frequently during the development stage (and we have

come to quite like it). **This microcomputer has an ARM processor and 512 MBytes of RAM. On the Raspberry, we now achieve reaction speeds of under 3 seconds** (i.e. that's how long it takes to build up the interface or save a process). The **ArchivistaBox Dolder already achieves reaction times of between 1 and 2 seconds** maximum, and on **all other ArchivistaBox systems we are talking about significantly less than one second,** even for complex processes such as creating the annual balance sheet.

Secondly: 10,000 orders with the ArchivistaBox Dolder

We are currently adding test processes to ArchivistaERP. This reminded us of an old test for the ArchivistaBox Dolder. In the**last blog, we tested the scanning speed of the ArchivistaBox Dolder; this time (now with ArchivistaERP) we wanted to find out how long it would take to process around 10,000 orders.** If you look at the above code closely, you will already recognise the for-loop with its 10,000 iterations. We were curious to see what times the ArchivistaBox Dolder would score. Here are the test results:

```
real 5m42.823s   user 4m22.140s   sys 0m23.565s
```



Bestellung #	Kunde	Abteilung	Kundenbest. #	Bestelldatum	Benötigt am	
13343,	Committo AG	German Oppliger		28.08.2014	07.09.2014	German Oppliger
13342,	Committo AG	German Oppliger		28.08.2014	07.09.2014	German Oppliger
13341,	Committo AG	German Oppliger		28.08.2014	07.09.2014	German Oppliger
13340,	Committo AG	German Oppliger		28.08.2014	07.09.2014	German Oppliger
13339,	Committo AG	German Oppliger		28.08.2014	07.09.2014	German Oppliger
13338,	Committo AG	German Oppliger		28.08.2014	07.09.2014	German Oppliger
13337,	Committo AG	German Oppliger		28.08.2014	07.09.2014	German Oppliger
13336,	Committo AG	German Oppliger		28.08.2014	07.09.2014	German Oppliger
13335,	Committo AG	German Oppliger		28.08.2014	07.09.2014	German Oppliger
13334,	Committo AG	German Oppliger		28.08.2014	07.09.2014	German Oppliger
13333,	Committo AG	German Oppliger		28.08.2014	07.09.2014	German Oppliger
13332,	Committo AG	German Oppliger		28.08.2014	07.09.2014	German Oppliger

Effectively, the job required less than 5 minutes and 43 seconds; the ArchivistaBox Dolder thus achieves a factor of ten for processing orders as compared to scanning. Leaving aside the fact that 10,000 scans is really not bad, it should be noted that scanning involves more data by a factor of 100 or 1,000 – which also explains why scanning takes approx. 10 times longer. If we extrapolate the 5 minutes 43 seconds to 24 hours, the **ArchivistaBox Dolder can process a whopping 2.5 million orders per day** – not bad for a box with software and hardware coming in at CHF 360, or EUR 300.

A final comment: the ArchivistaBox Dolder contains 2 GBytes of RAM, 1.5 GBytes of which are used for the operating system located in RAM (which is not entirely insignificant here), so "only" 512 MBytes of RAM are available to the operating system. So much for 64 GBytes for several hundred invoices per day!

Prospects, availability and price from CHF 360 (incl. hardware)

ArchivistaERP has now reached a state where we feel we can recommend it unreservedly. In short, we enjoy it more and more each day and are happy to pass on this enjoyment. Of course, ERP projects cannot be handled without a certain amount of complexity and sufficient implementation work, but **only ArchivistaERP offers everything that ArchivistaDMS has been offering for ten years – a ready-to-use solution** that can be supplied completely pre-configured. And all this at a price of CHF 360, or EUR 300. (ArchivistaBox Dolder). Taking into consideration the set-up-costs and first monthly rate, you'll be hard put to even find an SaaS (Software-as-a-Service) offer for this price anywhere. And with an SaaS solution, you completely hand over data

sovereignty to the internet. It doesn’t have to be that way.
Thanks to the ArchivistaBox, **ArchivistaERP is available as a system to work with from the first project hour. The standard scope (sale, purchase, warehouse, production and accounting) is instantly available for work.** Thanks to the modular design with classes, ArchivistaERP can be adapted and/or enhanced to meet all conceivable requirements. You alone decide where, when and how ArchivistaERP is to be used. The **fact that you also receive ArchivistaDMS is merely an aside** here, as this is a matter of course for us.
And if you are now wondering how this can be possible at such a price, we kindly invite you to come and let yourself be convinced. Just a quick reminder: the **ArchivistaBox (DMS) has been available since 2005; ArchivistaVM has been around since 2009 – adding ArchivistaERP simply does not incur any additional work or costs, plain and simple.** Rather, thanks to ArchivistaERP, we are able to keep our processes significantly leaner than was or would be the case with other solutions. Forget about server rooms, data centres and clouds — a 600 gram fanless box with a few watts is enough to run your business.
As of now, ArchivistaERP 2014/IX is supplied with all our ArchivistaDMS boxes. In the near future, ArchivistaERP will also be released free of charge on the **ArchivistaCloud.** The further development of ArchivistaERP is currently progressing rapidly. Many things are just waiting to be realised. The best ideas for ArchivistaDMS and ArchivistaVM came from our customers, and the same will be true for ArchivistaERP. You let us know what your requirements are, and we handle the technical implementation. **We speak your language. When it comes to development, outsourcing is an alien concept to us. As an SME, we have now been at your service for more than 15 years** — and at your service alone!
What benefits does this have for me as an ArchivistaVM customer?
Above all: with ArchivistaERP, you no longer need to worry about whether the next release of the ERP software requires new servers and new software again. This may be bad for the sales figures of our ArchivistaVM servers, but it is nonetheless good for us: it allows us to use our existing ArchivistaVM servers for other projects and for a much longer time.
What benefits result for me as a DMS customer?
As an **ArchivistaDMS customer, you benefit doubly. Firstly, you can use ArchivistaERP free of charge** and, secondly, you benefit from the fact that any partial processes not mapped (perfectly) by the in-house ERP solution can be handled easily with ArchivistaERP. Furthermore, the development of ArchivistaDMS is optimally enriched by ArchivistaERP; examples for this include the adaptation efforts for Internet Explorer, which were integrated directly into ArchivistaDMS in the course of the ArchivistaERP development.
In addition, **fully-automated test runs for ArchivistaDMS customers will be available on every ArchivistaBox** in the future, as we have the necessary tools at our disposal — thanks to ArchivistaERP. Examples: ArchivistaERP creates an invoice and sends it to ArchivistaDMS, or a call-up from ArchivistaDMS can be tested with ArchivistaERP. Broadly speaking: ArchivistaERP provides the most powerful testing tool for ArchivistaDMS that ever existed and probably ever will exist. A part of this work on ArchivistaERP already flowed into the 2014/IX version of ArchivistaDMS. This means that with the ArchivistaBox 2014/IX, we are able to provide a decidedly stable DMS that has been comprehensively tested with all current web browsers.