Firebird White Paper

Risks of business software as a cloud solution

Holger Klemt, January 2026

What should you do if your Porsche won't start or your cloud ERP is no longer usable?

Some readers will no doubt have already read the recent articles about dissatisfied Porsche customers in Russia whose vehicles no longer start, because a globally implemented anti-theft device in the vehicle permanently blocks the vehicle as a precautionary measure due to a lack of cloud access and shuts off the fuel supply in such a way that it cannot be circumvented with simple tricks, even though some providers have already offered workarounds. Porsche has sanctioned all support for Russia for good reasons, so no solution should be expected from that side.

What does this story have to do with a cloud ERP solution?

Well, sooner or later, a number of customers will surely notice this as well.

The idea for this White Paper was triggered by the decision of a medium-sized German company that had been using a self-written Delphi/Firebird application at numerous locations and had expanded it over the course of the last 30 years, implementing processes specific to their needs. Not everything was perfect, but in the area of logistics in particular, constantly changing standards were implemented in such a way that they precisely met their requirements.

In recent years, limited in-house programming resources have been successfully supplemented by experienced external freelancers, to implement new processes on a project basis.

At some point, a new young IT employee joined the company. He had no programming experience himself and was still far from understanding all of the company's processes. His previous employer was a Microsoft system house, and the Microsoft standard solution Dynamics 365 sold there was, of course, his undisputed favourite because it can be used to do everything the company needs (which he still didn't have a complete understanding of, but anyway...).

Since the whole thing no longer requires any local hardware, because everything operates in the cloud, it is also incredibly future-proof. The two generations in the family-owned company's management team were apparently convinced by this argument that this was the only conceivable way forward.

The amount of around €500,000 for the service package estimated by the affiliated system house seemed acceptable to the managing directors. However, this did not include any recurring licensing and operating costs, which were added separately, and it was not entirely clear whether each barcode scanner would require an additional licence and whether the connection would meet the requirements of the product-specific logistics processes. I roughly estimated that the annual fixed costs for their locations and all relevant employees would certainly be in the region of €100,000.

It was also unclear whether the above-mentioned budget would not only import the master data deemed important by the system house, as usual, or whether complex call-off orders or volume contracts with historic data would even be available at all in the new software. However, it appeared that all orders would have to be re-entered and that old data would then be available in the old software.

How the system would generally handle rolled goods, which are sold by the metre but also sometimes recorded in centimetres in subsequent production, was still completely unclear, or at least none of the decision-makers who were asked about this could provide a fact-based answer.

It appears that the programming team either did not yet have the necessary version of the new ERP system for testing or did not consider testing to be relevant. It is particularly concerning that they did not check the basic functional requirements of the existing logistics hardware, which is crucial for the successful implementation of a new system.

I had the impression that they relied heavily on what I considered to be very general online documentation, which did not provide any answers to specific questions, but which allowed us to conclude, while investigating a possible API connection, that there was no publicly available API information on the topics we would normally need.

There was repeated talk of such data being stored somewhere, for example as CSV files, and that the system house could then easily solve the problem with better API documentation for partners, based on the presumed internal use.

However, one thing was immediately apparent: direct read and/or write access to all tables in the database was not possible at all, and even a complete export of certain content – and it was unclear whether this was even possible for all data – e.g. as CSV, could only be done a maximum of 10 times per month via the Cloud API. Due to the lack of a test environment, it was not even possible to see how long this would take. Somewhere in the small print, it also stated that even the number of API calls was very limited in the standard version. How the invoice would look if more were needed was not made clear anywhere.

What used to be common practice in the old application, i.e. quickly logging certain processes such as delivery note printing via triggers and then using a very simple proprietary program module to generate customer-specific QR code/data matrix labels with serial numbers or similar, is no longer an option.

During the online research process, it became apparent that there are partner system houses that can implement such requirements with additional products or separate projects. However, one thing became clear: doing it yourself is no longer an option.

When a major customer requests something like this, it takes days, weeks, months or even years with unclear budgets to implement. Or you end up doing things as they did 20 years ago, with extra Excel-based isolated applications, into which you then have to re-enter the production data that is already available from the ERP system when it is needed. According to my research, there are no ODBC drivers for this.

Well then, you shouldn't stop a traveller, and if the managing directors ultimately decide on this solution, it remains to be seen how long the old Delphi/Firebird-based software will continue to be the leading system. The schedule for the new software has the transition set for 1 January 2027, which I would say is quite ambitious...

But what would happen, for example, if in mid-2027 it became apparent that the new software did not come close to covering everything that the old software was capable of, and you then had to fall back on the old system, without re-entering all customer and production orders, time records, etc. into the old system? That wasn't really clear either.

According to my research, data export was possible a maximum of 10 times per month, but the scope and formats involved could not be determined anywhere, so the effort required to import the data into the old application is also unclear.

In the worst-case scenario, both systems will eventually be operated in parallel. Whether this justifies the time spent by employees on double data entry and training in the new system, especially considering the migration costs, which are likely to exceed €500,000, and the €100,000 in annual licensing fees, is a decision that must be made at a higher level.

Why am I sharing this customer story?

We have been carrying out individual customer projects for more than 25 years, and we are by no means limited to the Firebird world, even though this is often conveyed by software companies who, as IBExpert customers, use industry-specific standard software based on Firebird. With the help of ODBC drivers or Lazarus SQLDB native database access, we are very effective at mirroring important data between platforms such as Firebird, MSSQL, Oracle, MYSQL or Postgres, and then create simple solutions for standard software that does not allow this, such as generating the above-mentioned serial number barcodes or creating UPS or DHL shipping orders directly from delivery notes.

Since we usually access data in such projects directly via SQL using read-only access on any platform, these projects can be implemented swiftly.

However, if it requires not just a database user with a password and basic SQL knowledge to recognise the data structures, but also some kind of REST/API or other procedure, to query or even write back something that is hardly or not at all documented, then such a project requires a completely different budget calculation.

And if there is no interface available that end customers can use themselves, without external system programming and certainly without having to pay individually for each call, it may be that many processes, where manpower savings could be made, can unfortunately no longer be optimised.

A simple example: A very large corporation from Germany sends one of our customers in the CNC parts manufacturing sector requests for quotations in PDF format, the contents of which unfortunately cannot be reliably analysed with a PDF2TXT export because the column model is a complete mess. Unfortunately, such requests are often 150 to 300 pages long in A4 format with 200-500 items. The page breaks were then unfortunately always where they appear in the printout, etc.

In order to create a quote for which many product numbers may already have been created, it would now be standard practice for someone, who is certainly not just a simple clerk, to type in the 16-digit product numbers and compare the text with what is already in the ERP system, if that is the case. Unfortunately, details that seem trivial at first glance are often the reason why the product being offered or the quantity requested is decisive for quotation variations. How long the employee needs for this, and at what hourly wage, needs to be budgeted just for the quotation preparation alone as an additional overhead, before even a single euro is generated in sales, This is something each company must decide for itself.

Our customer receives several such enquiries each week. For this reason, we have implemented the following:

1. The PDF and the associated zip file are recorded as a new offer import in the software, i.e. simply read into BLOBS.
2. On the server side, a script recognises the new content and unpacks it into a path.
3. Using a suitable tool, all pages of the PDF are exported individually as graphics with a resolution of 300 DPI.
4. Using a suitable tool, only the areas between the header and footer are cropped from the graphics to match the template.
5. Using OCR, the part in between is then cleanly captured from each page and assembled into a text at the end.
6. From this text, items 00010, 00020, etc. are then recognised, recorded as a new quotation and automatically inserted as quotation items via SQL.
7. If details such as drawing or design files with subpaths are recognised from the ZIP file for this quotation item, these are automatically added as a preview BLOB in the ERP software so that the user can view them directly.
8. If the item number already exists in the item master data, quotation prices and, if applicable, order data are automatically added.
9. After importing the two files (PDF and ZIP), it takes less than 10 minutes to generate the complete quotation, even with well over 100 items, and the clerk can simply click on them one after the other and decide whether they should be offered.

Since we have read and write access to all relevant data in this environment, such a project can be completed in a few hours to a few man-days. However, if the entire relevant data can only be accessed cryptically via some API, the effort required for such a project cannot be estimated at all, not to mention the fact that sub-problems with BLOBS cannot be solved at all, which may make the entire capture impossible.

Is the widely proposed AI helpful for this?

Many cloud providers do not attempt to solve similar sub-problems manually, as we do, but instead refer to 'we use the AI function for this'....

Now imagine that the above enquiry documents include high-tech or armament details or design features that, as a supplier, you are allowed to securely upload somewhere in order to then determine the details using some engine running somewhere.

Whether a server in the USA, Europe or elsewhere takes over the AI part, with your own local database server instance, which also runs in a self-controlled colocation with its own server hardware and its own AI server, as a company you can then reliably identify the data storage locations to the end customer without any exceptions.

But if this is stored somewhere in the cloud by your ERP provider on database server-like instances that are not documented or analysable in detail, and which may then be used not only by one of its customers but by many end customers jointly, then this is a real grey area, regardless of any assurances made by the cloud operator, because with platforms such as Azure, you cannot even determine where this is actually running. Since the cloud platform may also have automated backup procedures and you don't really have any influence over this, even if you receive a request to delete all data relating to a customer, you cannot guarantee that you will be able to comply.

And now comes the real problem with agent-based AI integration, i.e. the solution used by the ERP cloud provider can deploy additional servers anywhere on the planet to analyse sensitive file content. I very much doubt that the AI server provider will only use the details identified from this just for this one customer and then remove them from the overall model. After all, Facebook has already come under criticism for apparently using private chat content etc. from Facebook posts or even WhatsApp and Instagram to train its AI models. The value of complete access to ERP database content from manufacturing companies can be set much higher, in order to perhaps respond much better to industry-specific queries, thereby attracting professional users with higher fees. It is not without reason that platforms such as ChatGPT like to search directly on platforms such as Delphipraxis.net for questions about Delphi and Pascal source codes, without asking any of the administrators whether they think this is a good idea. (Delphipraxis.net was really hardly usable until the operator Thomas was able to implement a suitable solution to prevent such BOT queries).

In addition to the risks described here, there is another, much more serious problem: what happens if the ERP cloud provider offers cooperation for 'economic' reasons, giving corporations access to your ERP data without informing you in any way, regardless of what was cryptically stated in the terms and conditions?

In this case, you at least still have access to your data, even if it seems that others may also have invisible access to your data - who knows?

However, if the system used by the ERP cloud provider is itself ever paralysed by an attack or completely shut down due to economic difficulties, such as unpaid bills due to insolvency, you will no longer have access to anything. I would say: a difficult situation...

But even in the case of unilateral price adjustments, which will certainly be price increases, especially for regular, completely dependent customers, you have no real basis for negotiation.

So, do not use AI?

No, not at all, it depends on the control of the AI instance. This is feasible on your own hardware. We are currently working on a completely isolated AI platform for a customer, i.e. a hardware server that will ultimately be used by the customer with an LLM and complete access to all data from its non-cloud-based ERP Firebird database, including all documents and data from the last 15 years for training purposes, without the need for any instance accessible via the internet. It should be ready for series production in the course of 2026, and we will report on this in due course.

Finally, here is a checklist for risk assessment that should be considered before switching to cloud ERP:

---

1. Business risk: dependence on a single provider

Switching providers later on becomes virtually impossible or extremely expensive because data, processes and integrations remain proprietary. The provider can change prices, contract terms or product strategy – and you have little bargaining power. Roadmap dependency: if important features are missing, you cannot add them cleanly.

---

2. Exit costs and migration risk skyrocket

Without a complete, locally recoverable backup, a change is usually not a 'move' but a complete rebuild. Typical consequences:

- Months of parallel operations (continue to pay for and use the old ERP, roll out and pay for the new one).
- Training costs for employees to learn the new solution.
- Data reconstruction from CSV/reports instead of a real database/object structure.
- High consulting costs (mapping, cleaning up, processing historical data).
- Result: TCO (total cost of ownership) ends up being significantly higher than planned.

---

3. Data sovereignty and operability are restricted

Not having your own backup means that you do not have a genuine copy of your business and accounting data in a form that you can use independently. In the worst case (dispute, termination, insolvency, account suspension):

- No access = business comes to a standstill (or is only partially operational).
- Evidence/auditability suffers because you cannot freely reproduce what was in the system and when.

---

4. Compliance, auditing and documentation requirements become more difficult

- Relevant for managers because: fines, liability, damage to reputation.
- If exports are incomplete (e.g. missing workflows, log data, documents, versioning, authorisations).
- Inspectors/auditors may request documentation that you cannot provide. Rules on keeping records (like audit-proof chains of evidence) can be broken in practice.

5. API limitations block growth and efficiency

A 'small API' is not just technology – it is a business capacity limitation:
- Integration with shops, warehouses, production, BI, DMS, ticketing, payment and logistics becomes expensive or impossible.
- Lack of automation (e.g. through workflows, event hooks, webhooks)? More manual work? More errors? Less scalability.
- Reporting/BI remains superficial because you cannot access the raw data/structures.

6. Process sovereignty is lost (customising only within the cage)

If customising is only possible 'in the system' but not cleanly via API/extensions:
- You adapt processes to the ERP (instead of the other way around).
- Specialist departments build workarounds (Excel, shadow IT).
- This increases operational risk and compliance risk and worsens turnaround times.

7. Data extraction is not the same as data portability

Many providers offer 'export', but:
- Only master data and bookings, not document images, histories, comments, attachments, workflows, authorisation models, audit logs.
- Not in a format that another ERP can fully import.
- No way to test the export (restore/replay)? Exit remains theoretical.

8. Dependence on availability & incident handling

The cloud is fundamentally acceptable – but without an exit/backup, you are completely at the mercy of others:
- Downtime = standstill (invoicing, purchasing, logistics, production).
- The provider decides on RTO/RPO (recovery time/point of recovery), not you.
- Manager's perspective: daily revenue risk + supply chain risk + customer satisfaction.

9. Security & access model: 'You only get what the provider allows'

Limited API often means limited options for:
- Monitoring connection
- Forensic evaluations
- Externally checking granular rights/role models
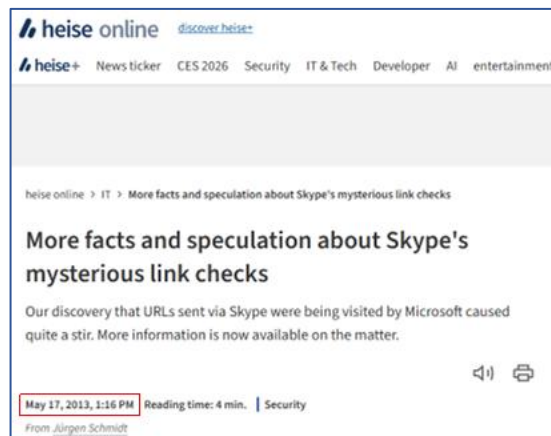- This increases the risk in the event of security incidents: lower transparency, longer response times.

> **10. Strategic risk: product changes, feature removal, M&A**
>
> The provider may:
> - Discontinue modules
> - Increase prices per user/transaction
> - Force you into a higher package
> - Change strategy following acquisition (M&A)
> - Without portability/backup, you are trapped

If you don't have a complete, locally usable backup and a sufficient API, you're not just buying software – you're buying dependency. And dependency is a financial risk, not an IT detail.

Do you believe that large, established corporations would never do such a thing?

Well then, here is a background story from 13 years ago, following Microsoft's acquisition of Skype: Skype's ominous link checks.

This is not the only reason why we have been using signal.org for internal company communication for a long time and are increasingly using it as a replacement for email when communicating with customers.

However, given the circumstances, you should be careful not to invite just anyone into a group chat on signal.org and then divulge American military secrets as the US Secretary of War. This is not a systematic problem with the signal.org platform, but rather with unqualified users.

And finally, a message to all those who continue to believe that their data will certainly not be misused by Office 365 and Outlook, which are owned by an American corporation with close ties to the government ...

IBExpert White Paper

Join our Signal Know-How Group

[IBExpertKnowHowSignalGroup](#)

Everything that is IBExpert and Firebird related. Please introduce yourself with the email that you also use in IBExpert Download Center to be added in the group. Preferred language: English.

Contact:
IBExpert GmbH
Oldenburger Strasse 233
26203 Wardenburg/ Germany
register@ibexpert.biz
www.ibexpert.com