



# White Paper

White Paper by Bloor  
Authors **Philip Howard**  
Publish date **May 2017**

---

## **Governance for Spreadsheet Models Apparity**

# Contents

Executive summary	3
Models	4
Basic requirements	5
Advanced requirements	6
Apparity	9
Summary	11
About the author	12

“

We particularly like some of the features offered by Apparity, which will suit companies that want version control, but also want the ability to save, send and edit spreadsheets unconstrained by designated folders or naming standards plus they want to do this without losing track of any copy of their managed spreadsheets, while identifying any and all changes made anywhere, at any time, to any spreadsheet version whether inside or outside of the firewall.

”

# Executive summary

**S**preadsheet Management is a broad subject. For some companies, it represents a compliance and governance challenge. For others, it is about risk management. For many, it is simply about ensuring that important spreadsheets are accurate and do not include errors, and that possibilities for fraud are eliminated. Yet again, it may be a case of supporting collaboration in an organised way, allowing multiple users to work together on a single spreadsheet. Or it may be about reducing the proliferation of spreadsheets or ensuring that users are always working with the current version of a spreadsheet, as opposed to some historic one. Or, of course, companies may be interested in a combination of these things.

Needless to say, with such a plethora of issues, different vendors in this space have sought to concentrate on diverse solution sets, with any one company's product more suited to resolving some spreadsheet issues as opposed to others.

Arguably the most complex spreadsheet environments occur when they are used to create and manage models. Most commonly, these occur in highly regulated industries such as banking, financial services, utilities, and large public companies subject to Sarbanes-Oxley and similar compliance obligations. We will discuss, in a moment, what we mean by "models" but these tend to be the cases where there is the greatest proliferation of spreadsheets, where spreadsheets are most mission critical, and where the largest (and most complex) individual spreadsheets are present. Once we have discussed what we mean by model-based spreadsheets, we will identify the issues that are involved with management and governance within a spreadsheet environment, and discuss best practices.

We should say that while this paper focuses on spreadsheets the technology we are discussing is also appropriate for governing other EUC (end user computing) resources, such as Access databases, PowerPoint presentations, pdf. documents, and so on. Given that the sort of models we are discussing – see next – often include or involve intellectual property, we are not here considering cloud-based solutions or anything other than Microsoft Excel.

Finally, note that this paper is in two parts. The first part is a reproduction of a recently published spotlight paper by Bloor Research on spreadsheet governance for model-based environments. Other than this paragraph and the conclusion at the end of this paper there are no changes between this paper and that original. The second part of this paper specifically discusses the spreadsheet solution offered by Apparity and this section will discuss how Apparity meets the requirements outlined in the first section. Readers will note that some of the capabilities discussed as requirements refer directly to capabilities offered by Apparity. We make no apology for this, as Apparity takes a markedly different approach, in a number of areas, compared to other vendors, and we believe that it is important to highlight this even in a generic white paper.



**Arguably the most complex spreadsheet environments occur when they are used to create and manage models.**



# Models

**I**n the context of this paper it is important to understand what we mean by a “*model*”. This is best illustrated by example. For example, you are a motor manufacturer and you have a pricing model that reflects all of the different options that are available for each of the different cars that you make. Or the pricing (and profitability) of each sale of your products or services or utility are dependent on many inputs, some of which (for instance, raw material prices) may fluctuate. Or you have an actuarial model. Or a financial model. Or perhaps your model is based more on engineering: for example, how best to configure wind turbines for power generation within a wind farm. Or your focus is more scientific: for instance, modelling drag across an aerofoil. Or you are modelling disaster scenarios for insurance purposes. Or, in the same industry, you are modelling your capital reserves, in order to ensure Solvency II compliance. Or you have similar requirements within banking, or for risk modelling. In addition, spreadsheets can be used to capture a variety of statistical or algorithmic models: for example, for algorithmic trading within capital markets. These are all types of model for which you might use spreadsheets.

There is also a second class of model, which can be instantiated within a spreadsheet, which can broadly be classed as business process models. For example, a budgeting application is an example of an automated business process which is often captured using a spreadsheet-based approach.



## Examples of spreadsheet-enabled models include:

- Pricing models
- Actuarial models
- Financial models
- Engineering models
- Scientific models
- Disaster models
- Capital reserve models
- Statistical models
- Algorithmic (trading) models
- Business process models



# Basic requirements

In this section, we are going to briefly discuss issues that are generic to all sorts of spreadsheet environments, regardless of whether these are model-oriented or not. We would expect all vendors in this market, regardless of their focus, to support all of the outlined requirements. These include:

- **Discovery** – the first thing that you need to do is to discover what spreadsheets exist and where they reside on the network. However, it is not enough to treat this as a one-off exercise because new spreadsheets are created all the time. This means that you need to discover new spreadsheets on an iterative basis. As a part of the discovery process you will also want to discover who owns the spreadsheet, who has access to it (and whether they actually use it), version information and details about links, both to other spreadsheets and to other resources such as databases.
- **Risk assessment** – once you know what spreadsheets you have, the next issue is to determine which are the most critical. If you want to take control of your spreadsheets then you will not usually be able to do this all at once so, you will have to prioritise which sheets are most important and which need to be taken under control, and this will be based on a risk assessment. The main elements within this risk assessment are threefold: complexity (the more complex the spreadsheet the more likely it is to break), materiality (how important to the business is this spreadsheet?), and frequency (how often is the spreadsheet used).
- **Error detection and correction** – a lot of the functionality required to detect and correct errors will be provided in the more recent versions of Excel but users of older versions may require specialised capabilities (for example, formula mapping, circular reference detection, precedent and dependent mapping, and so on).
- **Testing** – if you are building business critical applications based on spreadsheets, whether these are model-

based or not, then these applications should go through the same testing and quality assurance processes as any other application development project. Unfortunately, this is generally not the case (because these projects are run by users and not by IT) and, arguably, the failure to put spreadsheet applications through a formal testing process is the biggest cause of spreadsheet failures. Note that this is not a facility that any particular tool can provide: while support for testing might be included, your organisation needs to have a formal testing process. In the opinion of Bloor Research formal governance policies should be in place that mandate appropriate testing and approval processes prior to the release of spreadsheet-based applications.

One further feature that may be required in a generic sense is when automating manual business processes. Imagine, for example, that you are a distributor for a particular product and each month, each of your retailers sends you details of what they have sold. These details arrive in a variety of formats resulting in a business process that typically includes activities such as cut and paste, the inclusion of non-spreadsheet data (for example, from text files), frequent switching between files and spreadsheets, column and row manipulation, and so on. The problem with doing these things manually is that the process is not formalised, which means that it is prone to error and non-repeatable; there is no data lineage, so the processes cannot be compliant with regulations such as Sarbanes-Oxley; and it is non-trivial to undo what you just did if you make a mistake. What is required is tooling that can capture details of the manual processes that you go through each month, and then automate these so that the steps involved become reusable.

In the context of the previous paragraph one common error that occurs in model-based environments is that last month's spreadsheet is used instead of this month's. It is – or should be with appropriate tooling – simple enough to automate the process of ensuring that it is the current spreadsheet that is always used.



The main elements within this risk assessment are threefold: complexity (the more complex the spreadsheet the more likely it is to break), materiality (how important to the business is this spreadsheet?), and frequency (how many copies of the spreadsheet are there and how often is it used?).



# Advanced requirements

**I**n addition to the generic issues already discussed there are a number of other points that arise, especially (but not only) in model-based environments.

## Control

The idea of “*control*” of spreadsheets is to provide a management framework for the provisioning, development and versioning of spreadsheets and to ensure that users are only working with the latest version of relevant spreadsheets. All spreadsheet management applications provide this. However, “*control*” can also be applied to what users are allowed to do with spreadsheets. That is, can they change this value, amend that formula, and so on? In some environments, this sort of control may well be sensible. However, in others it is not. As a result, some vendors adopt an over-the-shoulder approach whereby they simply monitor what users do, while other suppliers take a more prescriptive line about what you can and can't do. Ideally, you like the option to do either (or both).

There are two ways to achieve this control. The historical approach has meant moving spreadsheets to, and hosting them in, a central repository or database. Care must be taken that any links to other spreadsheets or data sources are not broken during this process and relevant tools should offer the necessary facilities to ensure that these links are maintained. Once centralised, users log into that database to open spreadsheets that they want to use, make any relevant changes and then when they are saved a new version of the spreadsheet is automatically stored. Note that there is no spreadsheet to email to colleagues just a URL so that they can access it for themselves. You do, of course, need the ability to download a spreadsheet so that you can work on it offline with relevant check-in facilities when you log back into the central repository.

Alternatively, you can use “*fingerprinting*”. What this does is to put a fingerprint on each spreadsheet of interest, and that fingerprint remains

with all subsequent versions of that spreadsheet, which can be tracked by the governing software. It doesn't matter if the spreadsheet is renamed as something else, the fingerprint will still be there, which isn't necessarily the case when spreadsheets are stored in a database. A further advantage of fingerprinting is that if you have identified that two spreadsheets are linked then spreadsheets can still be tracked even when that link has broken. Without additional capabilities, you cannot do this when using a database-oriented approach.

There is a (theoretical) downside to using fingerprinting as opposed to a central database, which is that you cannot rationalise your spreadsheet real estate. When using a database, you typically store a single copy of each spreadsheet centrally, along with details of any changes that apply to versions of that spreadsheet. When you want to open a spreadsheet, you log onto the appropriate URL and, similarly, the idea is that if you want to share changes to your spreadsheet, then you share the URL with your colleagues rather than sending them a copy of the spreadsheet itself. Conversely, with fingerprinting, you cannot and do not attempt to prevent the copying of spreadsheets. Thus, the database approach potentially allows you to reduce spreadsheet sprawl. However, in practice, without stringent governance procedures, which most companies are reluctant to implement, this does not happen. So, this advantage is less than it might seem. Moreover, storage costs are so low that this aspect of rationalisation is not the issue it once was. The one place it might be an issue is where having multiple copies of a spreadsheet that contains sensitive data increases risk: here it might be necessary to impose strict governance policies and access controls.

## Disruption

It is worth expanding on the previous point. The big advantage of fingerprinting is that you don't have to move your existing spreadsheets



What you really don't want is to disrupt existing processes and procedures.



anywhere, which will be less disruptive and cost less. But regardless of how it is achieved, what you really don't want is to disrupt existing processes and procedures. You want users to continue to use their spreadsheets in exactly the same way that they did previously so that any change in the management and governance of those spreadsheets needs to be transparent.

It might – we emphasise “*might*” – be possible to achieve this even if you are storing those spreadsheets centrally and serving them centrally. However, that will necessarily imply a complex implementation process with a database to be installed, spreadsheets to be moved and so forth. So, there will certainly be some sort of disruption at the IT level, even though you might argue that IT is used to disruption.

An additional point about eliminating disruption is that spreadsheets sit within existing processes within the organisation and they are governed, or should be governed, by relevant policies. It is important, not least from a disruptive perspective, that any spreadsheet governance solutions should easily fit into, and work with, those existing process and policies. This will almost certainly mean that preferred solutions have strong workflow capabilities that can integrate directly into the corporate environment.

The bottom line is that disruption should be held to an absolute minimum.

### Performance

Another issue that can arise is with respect to performance and scalability. We have included this issue in this section because it tends to be the companies with model-based spreadsheets that run into this problem. We have met companies with hundreds of terabytes of spreadsheet data and we are aware of companies with individual spreadsheets that are ridiculously large (hundreds of megabytes). In such cases, you can get performance issues, depending on how your spreadsheet governance tool works. To understand this, it is necessary to appreciate that most spreadsheet

tools store spreadsheets – when they are centralised – in a relational database. The key question is how each individual spreadsheet is treated. If it is stored as an entity, and every time the spreadsheet is updated then the entire entity is updated, then you are going to have a potential performance issue. So, what most vendors do is to either store deltas or reverse deltas. In the case of deltas this means recording the changes but not the whole spreadsheet. The latest version of the spreadsheet is therefore a composite of the stored version of the spreadsheet plus subsequent deltas. For reverse deltas, the spreadsheet is always up-to-date and the reverse deltas let you see how the spreadsheet was previously. Neither of these approaches is ideal because you have to open the spreadsheet in order to update it. Note that, using a NoSQL database, while possibly less expensive than hosting spreadsheets on a relational database, would still potentially suffer from the effects of, ultimately, storing a spreadsheet as a single database object.

There is an alternative approach. This is to treat a spreadsheet not as an entity as such but as a string of XML code (which is what a spreadsheet is, under the hood). Now you can use source code management (SCM) software to manage change and versioning. When there is a change you update the XML in the live spreadsheet and store the reverse delta for audit trail purposes. This is certainly a more elegant solution, since this is exactly what SCM software is designed for. It eliminates the need for licensing a database to host your spreadsheets and, especially as the size and number of spreadsheets grows – and the activity thereon – we would expect improved performance from using SCM as the basis of a solution.

### Security and compliance

As we have noted, model-oriented spreadsheet environments often involve intellectual property that needs to be protected. Cell level protection with role-based access and integration with



**It is important, not least from a disruptive perspective, that any spreadsheet governance solutions should easily fit into, and work with, existing processes and policies.**



tools such as Active Directory and LDAP are likely to be required. More broadly, if Excel is being used for any sort of purpose that includes sensitive data about individuals, then you will need to be able to mask (pseudonymise) that data, in order to comply with legislation such as the EU's GDPR (general data protection regulation).

From a compliance perspective, your spreadsheet governance environment should generate appropriate reports and dashboards that can provide information about the relevant state of compliance. These, and more general management reports, should support drill-down into fine levels of detail. Considering governance more widely, it will be useful if your software integrates with broader data governance solutions from the likes of IBM, Informatica, Oracle and so forth: you would rather have one governance dashboard than several and the same applies to compliance. In practice, this probably means that you would expect your spreadsheet management supplier to provide a (RESTful) API so that the various environments can interoperate.

### Operational features

Finally, there are a range of operational requirements that are not really specific to spreadsheet environments, but which apply here as much as anywhere else. These include ease of use, self-service capabilities for the business, workflow (already mentioned) to support collaborative activities, and support for real-time as opposed to batch processing. This last requirement especially applies, in this case, to change management and, as a particular case in point, to the ability to distinguish between trivial (insignificant) and non-trivial (significant) changes. To explain this further, what we mean here is – for example – that if you change a formula then that is significant. However, if you restructure a spreadsheet by adding or deleting rows or columns this may appear to change formulae, but is not actually doing so. If you need to report on spreadsheet changes for compliance reasons (Sarbanes-Oxley et al) then it is important to be able to distinguish between these trivial and non-trivial changes because confusing the two can lead to a lot of additional and unnecessary work.



**This last requirement especially applies to change management and the ability to distinguish between trivial (insignificant) and non-trivial (significant) changes.**



# Apparity

**A**pparity is a second generation spreadsheet governance vendor that specialises in the management and “control” of spreadsheets within highly regulated industries, especially banking, insurance and the utility sector, but also in other areas where there are significant compliance requirements. The following provides brief details of the Apparity product with respect to the requirements previously outlined herein. The company’s solution is typically deployed on premises (behind the firewall) but is also available as a cloud-based solution. The company uses a subscription-based licensing model.

- **Discovery.** You can run discovery in Apparity either locally on each server independently and incrementally (that is, as required), or you can do it en masse from a central location. Scheduling facilities are provided to support ongoing discovery. A notable feature is that this discovery process groups the results it presents by spreadsheet version, regardless of the name associated with the spreadsheet. Note that this is prior to fingerprinting (see below) and is based on pattern recognition capabilities built into the product. This allows you to identify and, if required, remove duplicate spreadsheets. Further features of the discovery process are to find out who owns each spreadsheet and to discover the links that exist between spreadsheets and external sources. With respect to links Apparity is just about (at the time of writing) to release Model Map Explorer, which will allow you to graphically explore those linkages.
- **Risk Assessment.** This process works in Apparity pretty much the same way that it works with other vendors in the sense of applying risk factors to determine criticality. These risk factors are user defined and may be based on content (importance), fragility (complexity and links), reuse (a large number of copies increases risk) and/or any combination of these factors. Some spreadsheets may simply be designated as high risk, without going through a risk assessment process.
- **Error detection and correction.** As mentioned in the text, this is really only a requirement for users of older versions of Excel. Microsoft acquired Prodiance (a spreadsheet management vendor) in 2011 and has subsequently built that company’s error detection/correction facilities into later versions of Excel. Apparity pre-dates this acquisition so built what are now fairly standard features for identifying and handling errors, which can be used if you are using an older version of Excel.
- **Testing.** This is an area that is largely ignored by vendors. In fact, Apparity does offer some tooling to support testing but that is not really the point. The real point is to have testing and quality assurance processes in place, of which formal testing plays a part. The problem typically is that users building spreadsheet models do not have the discipline of formal testing procedures that software developers, for example, would deploy as a matter of course.
- **The automation of manual processes.** Apparity has built-in workflow capabilities that allow processes to be captured and automated. It is notable that by no means all of the company’s competitors offer such functionality. It is also worth bearing in mind that workflow can be used in all sorts of contexts: to assist with collaboration, for formal approval and release processes and even, bearing in mind the previous paragraph, to mandate testing as a part of an approval process.
- **Control.** Unlike most of its competitors Apparity does not move spreadsheets into a central location. Instead it uses the fingerprinting discussed previously. We have already discussed this in detail and we will not go back over that ground, though we should mention that fingerprints still apply even if, for example, you copy a spreadsheet

“  
Users building spreadsheet models do not have the discipline of formal testing procedures that software developers, for example, would deploy as a matter of course.

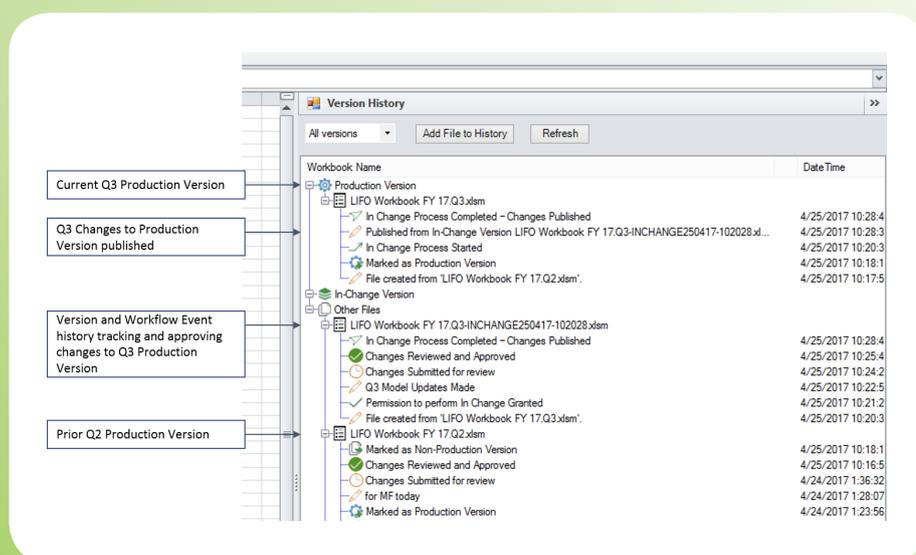
”



The use of SCM by Apparity means that its change log understands structural changes in a spreadsheet so that significant and insignificant changes can easily be distinguished from one another.



Figure 1: Apparity Change Management



into a Word document. However, it is worth commenting that in addition to role-based access control and authentication (see operational requirements), Apparity also implements Digital Rights Management (DRM) so that you can implement the sort of governance policies that are required to prevent unnecessary copying of spreadsheets.

- **Disruption.** As we have mentioned, Apparity does not move your spreadsheets, change the way you open or access them (except for the possible imposition of DRM), or centralise them in any way. There is no need to change user behaviour. The only data that Apparity does collect is metadata, which will be invisible to users. Thus, there is very little disruption involved in implementing Apparity. Indeed, this is an aspect of the product that the company focuses on. We should add that you define which spreadsheets are to be “managed”; other spreadsheets are not affected at all.
- **Performance.** Apparity uses Apache Subversion (an Apache open source project) for source code management (SCM) as discussed above, and does not rely on a relational database.

- **Security and compliance.** As noted, only some spreadsheets will be “managed”. Where this is the case single sign-on is implemented along with role-based (users, user groups and roles) security. There is support for Active Directory, LDAP and other approaches for authentication. Password protection can be applied to distribution. As mentioned, DRM is also provided. From a compliance and reporting perspective various features are provided, including audit, compliance reporting and so forth. Notably, functions such as comparing spreadsheets can be performed in real-time: main competitive products are limited to batch-based reporting.
- **Operational requirements.** There are a variety of operational requirements some of which, such as workflow, have already been discussed. The most important of those that have not is change management. This is comprehensive in Apparity, with all changes being tracked and the relevant metadata stored in the Apparity repository. The following annotated screenshot (Figure 1) illustrates some of the capabilities that are provided. As can be seen, this provides review and approval capabilities (via workflow) and distinguishes between production and other versions of a workbook. The use of SCM by Apparity means that its change log understands structural changes in a spreadsheet (row/column inserts/deletes) so that significant and insignificant changes can easily be distinguished from one another. Finally, it is worth adding that you can configure the software so that when a user submits a change for approval, the manager will be notified if the user has breached any policy directives (which can be defined and managed within the product) in the process of generating that change.

# Summary

**B**loor Research has been writing about spreadsheets and their governance for more than a decade. In that time, the vendors and the market for spreadsheet governance have evolved. For example, ten years ago there was very little in Excel to support error checking. That has now changed so, except for companies using very old versions of Excel, this is not a major issue anymore. That has meant that other considerations – especially “soft” requirements such as lack of disruption, collaboration and self-service – have come to the fore, as well as issues such as performance and scalability. We have discussed these, and other requirements, in detail in this paper.

From a vendor perspective, what we are now starting to see in this market is “*second mover advantage*” with newer entrants to the market – such as Apparity – adopting innovative and (more) modern technologies compared to suppliers with a longer history, that are (or may be) constrained by earlier technology choices. This does

not mean – of course – that older market incumbents cannot meet the requirements we have outlined, but they may find it more difficult to do so. As a result, we recommend that users considering the implementation of spreadsheet governance software, should consider a wide range of potential suppliers and not limit themselves to the more well-known providers. We particularly like some of the features offered by Apparity, which will suit companies that want version control, but also want the ability to save, send and edit spreadsheets unconstrained by designated folders or naming standards – which is what you get with relational databases – plus they want to do this without losing track of any copy of their managed spreadsheets, while identifying any and all changes made anywhere, at any time, to any spreadsheet version whether inside or outside of the firewall. Meeting that set of requirements is pretty impressive.



**From a vendor perspective, what we are now starting to see in this market is “*second mover advantage*” with newer entrants to the market – such as Apparity – adopting innovative and (more) modern technologies.**



### FURTHER INFORMATION

Further information about this subject is available from [www.bloorresearch.com/update/2331](http://www.bloorresearch.com/update/2331)



### About the author

**PHILIP HOWARD**

**Research Director / Information Management**

**P**hilip started in the computer industry way back in 1973 and has variously worked as a systems analyst, programmer and salesperson, as well as in marketing and product management, for a variety of companies including GEC Marconi, GPT, Philips Data Systems, Raytheon and NCR.

After a quarter of a century of not being his own boss Philip set up his own company in 1992 and his first client was Bloor Research (then ButlerBloor), with Philip working for the company as an associate analyst. His relationship with Bloor Research has continued since that time and he is now Research Director, focused on Information Management.

Information management includes anything that refers to the management, movement, governance and storage of data, as well as access to and analysis of that data. It involves diverse technologies that include (but are not limited to) databases and data warehousing, data integration, data quality, master data management, data governance, data migration, metadata management, and data preparation and analytics.

In addition to the numerous reports Philip has written on behalf of Bloor Research, Philip also contributes regularly to *IT-Director.com* and *IT-Analysis.com* and was previously editor of both *Application Development News* and *Operating System News* on behalf of Cambridge Market Intelligence (CMI). He has also contributed to various magazines and written a number of reports published by companies such as CMI and The Financial Times. Philip speaks regularly at conferences and other events throughout Europe and North America.

Away from work, Philip's primary leisure activities are canal boats, skiing, playing Bridge (at which he is a Life Master), and dining out.

## Bloor overview

Technology is enabling rapid business evolution. The opportunities are immense but if you do not adapt then you will not survive. So in the age of Mutable business Evolution is Essential to your success.

***We'll show you the future and help you deliver it.***

Bloor brings fresh technological thinking to help you navigate complex business situations, converting challenges into new opportunities for real growth, profitability and impact.

We provide actionable strategic insight through our innovative independent technology research, advisory and consulting services. We assist companies throughout their transformation journeys to stay relevant, bringing fresh thinking to complex business situations and turning challenges into new opportunities for real growth and profitability.

For over 25 years, Bloor has assisted companies to intelligently evolve: by embracing technology to adjust their strategies and achieve the best possible outcomes. At Bloor, we will help you challenge assumptions to consistently improve and succeed.

## Copyright and disclaimer

This document is copyright © 2017 Bloor. No part of this publication may be reproduced by any method whatsoever without the prior consent of Bloor Research.

Due to the nature of this material, numerous hardware and software products have been mentioned by name. In the majority, if not all, of the cases, these product names are claimed as trademarks by the companies that manufacture the products. It is not Bloor Research's intent to claim these names or trademarks as our own. Likewise, company logos, graphics or screen shots have been reproduced with the consent of the owner and are subject to that owner's copyright.

Whilst every care has been taken in the preparation of this document to ensure that the information is correct, the publishers cannot accept responsibility for any errors or omissions.



Bloor Research International Ltd  
20-22 Wenlock Road  
LONDON N1 7GU  
United Kingdom

Tel: **+44 (0)20 7043 9750**  
Web: **[www.Bloor.eu](http://www.Bloor.eu)**  
Email: **[info@Bloor.eu](mailto:info@Bloor.eu)**