

# White Paper



## Usage-based billing as a database platform service

Pervasive UseaBiLLity metering at the database level for ISV value-add billing and usage analytics

Bloor can see no reason why the services model, and usage-based billing, won't rapidly become the norm for software acquisition

David Norfolk

## Executive summary

Information technology is moving to a services model: technology is being rented rather than bought outright and, in this world, customers want to pay only for what they did use rather than for what they might want to use. Of course, if they are only renting technology, the barriers to moving to a different supplier, if they are unhappy with a supplier relationship, are much reduced.

This will affect the business models for the ISV (independent software vendor), VAR (value added reseller) and OEM (original equipment manufacturer) solution vendor; as well as that of the people supplying the technology platform underneath. People have to make money somehow and the traditional software licensing model is rapidly becoming obsolescent.

The answer is to find chargeable services that customers want, and bill for them in such a way that the customer perceives the service benefit rather than the cost. This means transparent billing based on verifiable usage analytics, which is automated (with low overheads) and is associated with some form of value-add. Billing is becoming seen as a core component of the services model—see, for example, Forrester's PaaS (platform as a service) reference architecture model (Figure 2).

Fortunately, technology already exists to deal with this issue for many software services; those based around widely used and commoditised database systems with robust usage-based licensing mechanisms. 'Licensing' can be re-purposed as 'billing' and the database platform can supply, as a service, the information an ISV needs to produce a transparently fair billing application. This billing application can supply the ISV's customers, again as a service, with the usage information the customer can use to implement internal charge-back and to predict and influence end-user behaviour (people can be persuaded to minimise even notional charges, if they are seen as fair and accurate)—a value-add service.

As usual, the devil is in the implementation details, but instrumenting MySQL, in particular, for data metering is becoming a very practical possibility. Many ISV SaaS (Software as a Service) offerings use the MySQL database management system and this has recently introduced plug-ins and 'pluggable storage engine' features, which allows the addition of code to the MySQL server that can monitor real database accesses and events (a very

solid basis for billing) without affecting the server as seen by the applications using it, and without affecting MySQL licensing (see Figure 1). One example of such an engine can be extracted from Pervasive PSQL, which already has robust usage-based licensing and embedded capabilities and is slightly more efficient at accessing storage than the default engine supplied with MySQL (thus eliminating, in effect, any billing overheads). An important aspect of this engine is that it is tamperproof—encryption is used to ensure that no-one can corrupt the usage information it collects and processes. Of course, other engines could be written to exploit the MySQL architecture; but the intellectual property behind Pervasive's licensing/billing approach is protected by patent and has been tested in the marketplace.

Nevertheless, although the enabling technology is important (it must be robust, reliable and accountable), people should always remember that usage-based billing is really a business service. It is supported by database-level event and access metering but, most significantly, it should provide mapping from real-world technology events to business metrics that end-users understand (and therefore have confidence in), and that management can use for decision support and process improvement. It should be used, to deliver real and useful business benefits:

- Fair and tamper-proof billing for all involved.
- Business usage analytics that allow the business to make better decisions and to fine-tune desired behaviours.

### Fast facts

- Transparent billing is a fundamental part of an effective SaaS solution and the information associated with this can be used for customer-level (end-user) usage analytics and charge-back mechanisms.
- MySQL database is widely used and MySQL, early on, introduced a plug-in storage engine architecture that allows the addition of alternative storage engines, together with extensions such as usage-metering capabilities, to a running MySQL server without recompiling the server itself.
- The Pervasive UseaBiLLity plug-in provides an example of such a pluggable extension which, as a development of its licensing

## Executive summary

---

technology, supports a metering service with rich reporting of tamper-proofed usage information for use in external billing applications and end-user analytics.

### The bottom line

In the SaaS world, billing isn't an optional add-on; it is fundamental to the SaaS relationship—and that could be vital to the operation of your business. If, today, “the business runs on software”, tomorrow, the business (whilst still running on software) will really run on SaaS relationships—and if you are renting SaaS, nothing is easier than changing your supplier if the relationship goes sour.

What will sour a relationship is billing that isn't transparent, isn't trusted, isn't effective and which involves the customer in maintenance and technology overheads on top of their ISV's bill. This is an emerging issue for the ISVs and VARs that, increasingly, have to provide SaaS solutions to their customers but still need to make a living.

Instrumenting MySQL for data metering in such applications provides one real opportunity for SaaS vendors. The MySQL architecture provides a way to add reliable tamper-proof billing as a service to database applications, which then provides a value-add for ISVs and a valuable usage-based analytics capability to their customers—it also enables palpably fair billing, in which all stakeholders can have real confidence.

## Utility billing in detail

### The billing services opportunity

The opportunity provided by utility, database-enabled, usage-based billing is that it helps ISVs to 'SaaS-ify' legacy applications by adding to them robust, reliable and secure services that deliver billing and business-oriented metrics. We see usage data being used for predictive provisioning, price modelling—and for a recursive pricing, billing, provisioning process, throughout which the customer is looking both backwards (“have we been paying a fair price for this service?”) and forwards (“will we have enough capacity available?”); as well as for billing.

The essential requirements for any technology that delivers this opportunity are:

- No code changes are needed for existing applications to exploit this opportunity.
- Any database application is able to take advantage of this opportunity.
- No database license changes are needed.
- The mapping of database events and accesses to business-level activities that people can understand, pay for and manage, involves a one-time effort, based on existing metadata schemas and the ISV's existing application domain and business model knowledge.

- The technology is available as a service to ISVs and is also available as a software appliance (for OEMs).
- The technology solution supports any private cloud initiatives of the ISV or OEM.
- The technology solution is loosely coupled to business systems (that is, billing service/system agnostic) and integrates with and supports SAM (Software Asset Management) and IT Service Management initiatives.
- The technology solution provides full event logging per server and per application and thus provides robust support for rich business-level analytics.

The most obvious enabling technology for exploiting this opportunity is the MySQL architecture (Figure 1), which is what is exploited by Pervasive for its UseaBiLLity metering service. The current MySQL architecture allows you to simply drop a billing-enabled plug-in into the MySQL server, connect to a mapping service (with mapping provided by the ISV) and then forward usage metrics to a billing system for processing.

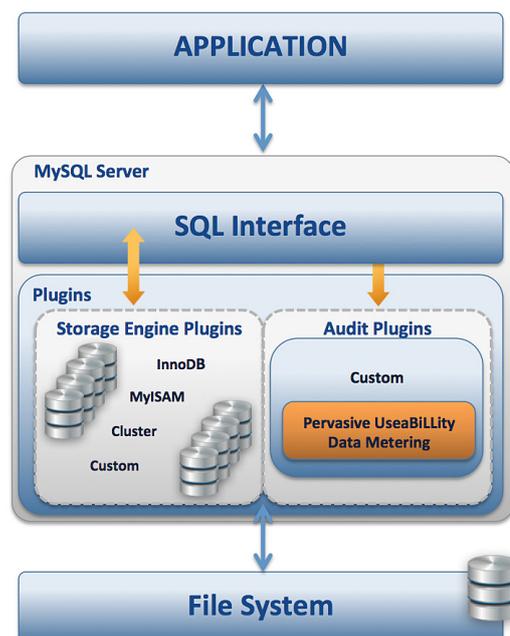


Figure 1: MySQL pluggable storage-engine architecture

## Utility billing in detail

### SaaS billing implications

Usage-based billing is a technology-enabled opportunity, but it is first and foremost a business story. ISVs sell software and need to make a living from it. The trouble is that cloud services and SaaS models are effectively rendering conventional licensing obsolete. Why would anyone want to take out a license that costs money even if you don't use the software? No-one, when the alternative is to pay for only what you use, when you use it—it is attractive to think in terms of a 'mobile phone tariffing' model, although the obfuscation associated with phone tariffs (leading to many phone users paying for more expensive tariffs than they need) should be avoided.

So, the SaaS vendor needs to bill for its software usage and, if you think about it, monitoring software licensing (which the more sophisticated software products already do) and billing are different aspects of the same problem. It is worth noting that the Forrester Platform as a Service (PaaS) reference architecture model includes billing as a key component (see Figure 2). However, as Phil Wainwright pointed out on ZDNET as far back as 2008 (<http://www.zdnet.com/blog/saas/the-future-of-saas-billing/484>), SaaS billing needs to be automated, but by something rather more sophisticated than an Excel spreadsheet filled in manually with raw data, often with a poor provenance.

SaaS billing must be based on real usage information, that can be understood and accepted by the people paying the bills; the stakeholders in selling, using and paying for software-based services must have confidence in billing, and assurance that the usage information it's based on can't be tampered with. However, this doesn't mean that SaaS vendors or their customers either want to, or should need to, write complex technical hooks into SaaS systems to capture usage; and neither will they want to be made to write billing applications that are technically more complex than the business can understand.

Also, of course, since many SaaS customers see SaaS as an opportunity to reduce unnecessary technology spend, SaaS vendors need extra value-add services, seen as useful by their customers, in order to maintain revenue.

If you can successfully re-purpose licensing logic that is already embodied in a product for SaaS billing, then you can provide benefits to all stakeholders in the SaaS relationship:

- The software OEM vendor gets additional income from its billing-enabling service.
- Its ISV gains a reliable income stream from its billing without upsetting its relationship with its customers.
- The ISV gets a service it can sell on, supplying management information to its customers.
- The ISV's customers get usage data for analytics allowing them to manage their technology spend, identify coverage holes, modify usage-related behaviours and so on.

This, of course, assumes that these billing-related services are transparent, tamper-proof, easy to make use of and have minimal overheads, both in terms of resource utilisation and licence/relationship management.

In essence, the SaaS model involves the implementation of a different way of managing customer relationships; often, traditional ERP tools and, especially, legacy billing applications simply aren't appropriate. What customers need today is confidence that they actually used a service that they paid for, and they want sufficient information on usage to allow them to implement internal charge-back. In a SaaS world, the people who use the technology should pay for it and be able to account to

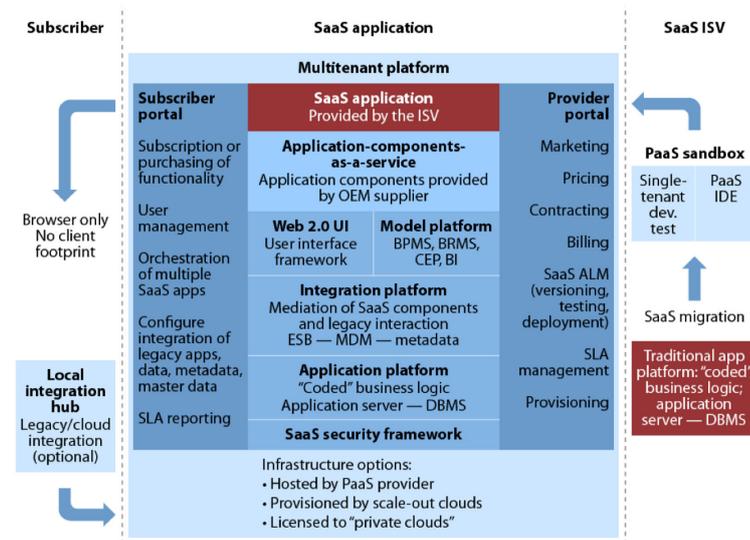


Figure 2: Forrester PaaS reference architecture model  
 Source: Forrester Research, Inc., The Forrester Wave™: Platform-As-A-Service For Vendor Strategy Professionals, Q2 2011, May 2011

## Utility billing in detail

---

their organisation for this, in terms of business outcomes. The organisation needs to be able to identify workers who are not using technology that would make them more productive and provide them with encouragement and training; so to modify their behaviour in desirable ways. For example, you could discourage undesirable behaviour by using charge-back to make it more costly, as long as the charge-back is seen as fair and honest.

The SaaS model is the enemy of the world in which an organisation pays for 10,000 SupaKollaboration licenses no-one uses—and another 500, more expensive, Enterprise SupaKollaboration Pro licenses for the one team that has actually worked out how to use this stuff and jealously keeps this a secret from the rest of the organisation. Detailed, utility-based billing built into the platform, providing transparent management information on usage and usage patterns (in a form that can go into a spreadsheet, if necessary), is an enabler for a better way of managing technology in the context of business outcomes and, therefore, is a service people will pay for.

### Vendors

There are two vendors involved in the billing services solution covered by this paper:

- MySQL is a market-leading open source database management system much used for web applications. It has some technical issues compared to, say, IBM's commercial DB2 but it is fit-for-purpose for most web applications, being actively developed, and likely to continue to be popular. MySQL has been acquired by Oracle, the market-leading commercial DBMS vendor, but Oracle appears to be leaving MySQL's open-source business model alone, and adds a measure of confidence in MySQL's commercial resilience.
- Pervasive is a long-established and profitable vendor of database technology (its PSQL database started off life as Btrieve), particularly in the embedded and OEM marketplaces.

## Pervasive UseaBiLLity utility billing

### Implementation of the service

Pervasive UseaBiLLity is a metering service, which is loosely coupled to the application platform. You simply drop the Pervasive plug-in, which allows you to add event-capturing capabilities to a running MySQL server, into MySQL Server (any version later than MySQL 5.1) and connect it to the Pervasive mapping service. A MySQL plug-in allows you to create new storage engines and add them to a running MySQL server without recompiling the server itself, and is fully described on the MySQL website at <http://mysql.netvisao.pt/doc/internals/en/custom-engine.html>.

The mapping service logs database events (the reasonable assumption being made that the work done by an application—its usage—relates directly to retrieval and storage of data):

- Data being input and output.
- Table Insert, Update, Delete, Read actions.
- Stored procedures being invoked.
- Transactions starting and finishing.

These are metrics for 'data in use' (billing for data that is just there but not being used seems unfair); but they don't mean much to the business people that pay the bills. One key feature of a truly useful mapping service is that it translates database events into business-oriented activities and forwards them to a billing system for processing, as well as making them available to end users for analysis of who is using what resources. Another key feature is prevention of data tampering, by encrypting the usage data.

Pervasive's UseaBiLLity service supplies statements of usage that can not only be forwarded to a billing service (either in-house or something outsourced like Zuora) but also used to predict future behaviours or events ("you are about to exceed your allocated budget because.....") or explain past ones ("you have exceeded your allocated budget because..."). This gives all stakeholders better control over negotiations and improves ISV-customer relations; as well as facilitating capacity planning and trend analysis.

A possible limitation of the approach described is that it only works with MySQL applications; but there are a lot of these out there and MySQL is popular with ISVs. Another possible issue is that not all applications are database usage-centric. It will probably not bill effectively for compute-intensive applications that don't store or read much data; nor for applications that store data for long periods (for audit trails and compliance, perhaps) but which seldom access or process it. Nevertheless, most business applications are data-centric and move data; and a lot of ISV offerings are built around MySQL.

### Service architecture

The architecture adopted by Pervasive for its UseaBiLLity metering service is described in Figure 3. UseaBiLLity encompasses a MySQL plug-in, which collects the usage data, and the Billing/Analytics Service, which translates database events into logical business activities that mean something to the business.

## Pervasive UseaBiLLity utility billing

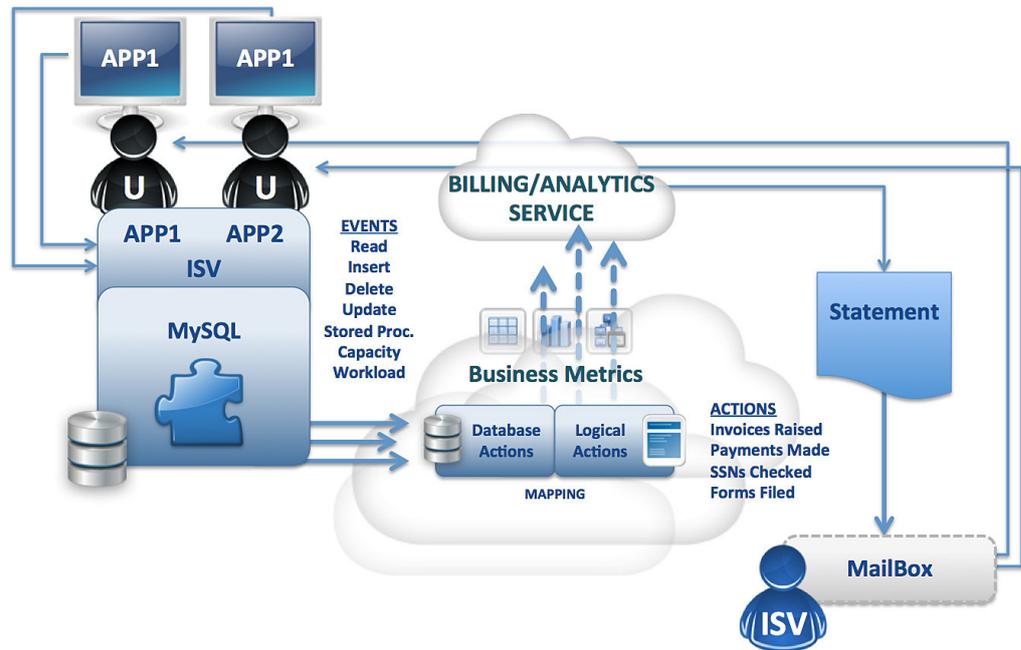


Figure 3: Pervasive UseaBiLLity architecture

### Differentiators

- This is a SaaS-oriented solution, not a crude update of a conventional licensing model. It is available as a service for ISVs (or as a software appliance for OEM vendors). The UseaBiLLity service can be consumed by any other service and is not limited only to Pervasive offerings.
- No code changes are required, either for ISVs or end users; and as the Pervasive plug-in is more efficient than the standard MySQL storage code, billing overheads are minimal.
- Any MySQL application can be enabled with billing.
- No MySQL license changes are required (customised storage engines are now a standard MySQL feature) and there is no SAM overhead.
- One-time mapping effort by the ISV (using its intimate knowledge of its MySQL application) means that billing brings no continuing maintenance overhead.
- Loose coupling (as a service) to the business application means that it is billing (and analytics) service agnostic—and can't impact business system reliability.
- All usage data is encrypted while in the hosted metering service, so that it cannot be tampered with.
- Choice of a hosted service (for ISVs), hosted by Pervasive, or a software appliance (for OEM vendors) is provided.

## Pervasive UseaBiLLity utility billing

**Title:** Fair “no surprises” usage-based billing.

**Actors:** Software services vendor and its customer.

**Scope:** Externally-sourced software services.

**Level:** User goal.

**Body:** Software usage is measured over a period of time and this is used to predict behaviour (and, thus, software usage and anticipated charges) over the subsequent period. This information can be used as a basis for negotiation to correct under- or over-payment in the measurement period and determine, for the subsequent period, a fair flat-rate rental charge; or a basic-rate fixed rental fee with anticipated ad-hoc charges for random peak usage; or a pure ‘pay per usage’ tariff with no surprises for either party.

### Customer Use Case

Many SaaS customers experience a ‘Billing Paradox’: they love the idea of only paying for what they use but, at the same time, budgeting and planning is easier if they pay a fixed price each month. This doesn’t make things any easier for an ISV trying to sell at a fair price.

This is where a billing service that makes the data it captures available for analytics comes into its own. Analysis of the behaviour represented by 6–12 months of actual usage data can predict the expected usage-based billing in future and, thus, help both ISV and customer agree on a fair, fixed, monthly charge; renegotiated at period-end in the light of actual usage in the previous period.

This is a powerful use case because it supports a real world continuum from purely fixed pricing to entirely variable usage-based pricing.

This use case can address the issue with, for example, many mobile phone tariffs where the customer often ends up on a higher tariff than they need, because they are asked to anticipate their behaviour without hard information to base this on (remembering that their past behaviour is likely to change if they are buying a new phone with improved features) and without the option to choose a tariff retrospectively. It exemplifies just one of the options that become available if a billing service provides trusted data analytics as a service, as well as just billing.

### Complementary products/services

Metering feeds any other applications that can use its data: billing, analytics, provisioning, etc. The output of these applications can, in turn, be consumed by others: so billing generates statements, invoices or charge-backs that need to be absorbed by ERP, asset management or even CRM (customer relationship management) systems (such as salesforce).

Pervasive’s other product offerings (such as Pervasive Data Integrator and even Pervasive RushAnalyzer for big data analytics) could play a part in this further processing. However, an essential aspect of its service-based approach to UseaBiLLity is that its metering services can be consumed anywhere they are appropriate, not just by Pervasive products.

## Summary

---

The SaaS services model for software, with 'Pay per use', 'Pay as you go', billing options, either pre- or post-paid and based on usage in business terms (that is, on verifiable business metrics) is replacing older subscription models (user- but not usage-based) and licensing models (based on perpetual maintenance/upgrade delivery and potential use, whether you actually use the software or not). Bloor can see no reason why the services model, and usage-based billing, won't rapidly become the norm for software acquisition: paying only for what you use from OPEX instead of CAPEX and being able to de-provision at minimal cost when you no longer need a service, is simply too attractive to the business to be ignored.

Billing is an essential part of the services model and if you can re-purpose database licensing logic to enable billing services and usage analytics—using database accesses, mapped into business activities, as a metric for usage—you can provide a service to ISVs that allows them to 'SaaS-ify' their existing applications at low cost, provide value-add input to end-user usage analytics, and better manage customer relations.

The new MySQL plug-in architecture and Pervasive's existing and tamper-proof licensing facilities provide the enabling technology for this style of billing. Bloor notes that the UseaBiLLity billing-services solution being promoted by Pervasive is limited to instrumenting MySQL for data metering and to database centric applications but, for many ISVs and end users, that is hardly much of a limitation in practice.

Bloor is of the opinion that usage-based billing services and provision usage analytics input as a service will be an important feature of the next-generation software-driven and service-oriented businesses and welcomes Pervasive's innovative approach. We hope this will serve as a model for the business generally.

### Further Information

Further information is available from <http://www.BloorResearch.com/update/2134>

## Bloor Research overview

---

Bloor Research is one of Europe's leading IT research, analysis and consultancy organisations. We explain how to bring greater Agility to corporate IT systems through the effective governance, management and leverage of Information. We have built a reputation for 'telling the right story' with independent, intelligent, well-articulated communications content and publications on all aspects of the ICT industry. We believe the objective of telling the right story is to:

- Describe the technology in context to its business value and the other systems and processes it interacts with.
- Understand how new and innovative technologies fit in with existing ICT investments.
- Look at the whole market and explain all the solutions available and how they can be more effectively evaluated.
- Filter "noise" and make it easier to find the additional information or news that supports both investment and implementation.
- Ensure all our content is available through the most appropriate channel.

Founded in 1989, we have spent over two decades distributing research and analysis to IT user and vendor organisations throughout the world via online subscriptions, tailored research services, events and consultancy projects. We are committed to turning our knowledge into business value for you.

## About the author

---

**David Norfolk**  
Practice Leader  
Focus Area: Development/Governance



David Norfolk first became interested in computers and programming quality in the 1970s, working in the Research School of Chemistry at the Australian National University. Here he discovered that computers could deliver misleading answers, even when programmed by very clever people, and was taught to program in FORTRAN. His ongoing interest in all things related to development has culminated in his joining Bloor in 2007 and taking on the development brief.

Development here refers especially to automated systems development. This covers technology including acronym-driven tools such as: Application Lifecycle Management (ALM), Integrated Development Environments (IDE), Model Driven Architecture (MDA), automated data analysis tools and metadata repositories, requirements modelling tools and so on. It also covers the processes behind them and the people issues associated with implementing them. Of particular interest is organisational maturity as a prerequisite for implementing effective (measured) process and ITIL (v3) as a framework for automated service delivery.

David is a past co-editor (and co-owner) of Application Development Advisor and associate editor for the launch of Register Developer, and is currently executive editor for GEE's "IT Policies and Procedures" product. He has an honours degree in Chemistry and is a Chartered IT Professional, has a somewhat rusty NetWare 5 CNE certification and is a full Member of the British Computer Society (where he is on the committee of the Configuration Management Specialist Group).

His early career involved working in database administration (DBA) and operations research for the Australian Public Service in Canberra. David then returned to his UK birthplace (1982) where he worked for Bank of America and Swiss Bank Corporation, at various times holding positions in DBA, systems development method and standards, internal control, network management, technology risk and even PC support. He was instrumental in introducing a formal systems development process for the Bank of America Global Banking product in Croydon.

In 1992 he started a new career as a professional writer and analyst. Since then he has written for many major computer magazines and various specialist titles around the world. He helped plan, document and photograph the CMMI Made Practical conference at the IoD, London in 2005 and has written many industry white papers and research reports including: IT Governance (for Thorogood), Online Banking (for FT Business Reports), Developing a Network Computing Strategy and Corporate Desktop Services (for Business Intelligence), the Business Implications of Adopting Object Technology (for Elan Publishing).

He has his own company, David Rhys Enterprises Ltd, which he runs from his home in Chippenham, where his spare moments (if any) are spent on photography, sailing and listening to music.

## Copyright & disclaimer

---

This document is copyright © 2012 Bloor Research. 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.



2nd Floor,  
145-157 St John Street  
LONDON,  
EC1V 4PY, United Kingdom

Tel: +44 (0)20 7043 9750  
Fax: +44 (0)20 7043 9748  
Web: [www.BloorResearch.com](http://www.BloorResearch.com)  
email: [info@BloorResearch.com](mailto:info@BloorResearch.com)