cube

Enabling Bespoke Embedded Business Insights Through Semantic Layers

A Robust Guide for Modern Embedded Analytics

Since my teens in 1980s Scotland I have been heads-down working with data in many different ways. Like many, my first practical data structures (linked arrays coded in BASIC or PASCAL) cataloged my books and music collections, but I was soon building databases for studies in history, archeology and language. I discovered PROLOG and LISP with delight. Later, in business, I harnessed data for efficiency in construction and agriculture. But working with large-scale systems in Scotland's oil and whisky sectors, I learned the distinct needs of operational and analytic frameworks: that these systems are conceptually and architecturally very different. You can't analyze data at enterprise scale on a system optimized for enterprise transaction processing. And so my passion developed for analytics as a practice.

Donald Farmer, February 2024

Business Intelligence: The Lost Opportunity

One of the most enjoyable phase-changes in this so-called career was in the late 90s when I moved from developing specialized data analytics services to building general purpose Business Intelligence (BI) tools. The aim in my own work, and then at Microsoft, and later again at Qlik, was to enable more people to take advantage of analytic insights: not just canned reports, but interactive experiences which allowed for at least some exploration and discovery even in the hands of business users.

You may remember the catchphrases from the time and the promises they held – BI for the masses, democratization of data and so on. Over 25 years later, I could sit back and think that effort succeeded. BI and analytics are now ubiquitous in business, and I played my part. But the truth is sobering: standalone BI tools have effectively reached a dead-end; even though almost every business uses some BI, adoption within enterprises remains undemocratically below 30%. What went wrong? And is there still a future for business analytics as we know it?

In this paper, I suggest that there is indeed a critical way in which BI can touch every worker – as Embedded Analytics integrated with operational applications. But we can go even further, enabling customer-facing analytics which deliver data and insight externally, embedded in the reports, applications or even invoices and statements that we share with customers and partners. As I look at the technical and business requirements for effective embedding, I will also describe why Cube in particular has some important advantages.

I also think there is a role for Business Intelligence as a complement to Artificial Intelligence.

First, let's review the aims of classic Business Intelligence originally and why, despite the commercial success of many BI vendors, the best opportunity to change the dynamics of business decision-making still lies ahead of us.

The Promise of Business Intelligence

Give me the right information, at the right time, in the right format...and I'll make the right decision

Sounds good? But BI was never quite so simple. The right information always represented a moving target, as business needs changed rapidly especially with increasing digitization and business processes moved online. This in turn drove a need for near-real-time decision making in many cases.

And the format? There were specific demands to be met, especially as users expected to be able to work on mobile devices. Meanwhile greater data literacy and a demand for better design meant that visualizations had to become ever more subtle aesthetically and yet ever more capable technically.

Behind all this, lay a question which BI vendors – my own teams included – never really answered in a satisfactory way: should analytics be a specialized job role or something which any business user can use?

The answer is still unsatisfactory. Reports, straightforward visualizations and simple

dashboards are now within reach of almost every business user. But those capabilities are easily offered by ERP vendors such as SAP, Oracle or Infor. Specialized BI vendors, such as Looker and Tableau, found they had to build communities of specialized users and did so very effectively. Somewhere in the middle, Microsoft created a huge market for PowerBI which, at the low end, proved somewhat better than Excel for analytics, and at the high end includes an expression language of intimidating complexity.

As a result, the promise of Business Intelligence – to improve decisions for every business user – has remained unfulfilled even though the developments in BI software have been considerable.

The Business Intelligence Workflow

For non-specialist users who still use traditional BI reports and dashboards, we generally find a lot of enthusiasm for analytics and insight at first, but then engagement gradually fades. It is not just the novelty wearing off, but often people come to realize that analytics interrupts their workflow rather than complementing it: **An interruption frequently not justified by results.**

A significant source of inefficiency arises when users must switch away from their operational applications – such as order processing or customer support systems.

Here's How:

- Initially, the user is engaged in an operational application. But when the need for analytical insight arises, the user must switch from an operational to an analytical environment, a process that disrupts their workflow and context.
- This involves exiting the operational application, logging into a separate analytical platform, and navigating a different interface to access specific reports or dashboards.
- After analyzing data, users must switch back to their operational tasks, facing yet another context shift.
- The data presented may not align directly with their operational context, requiring users to mentally connect it back to their tasks, a process that is cognitively demanding and prone to error.
- This back-and-forth can lead to analysis paralysis, especially if the analysis does provide valuable insight. Additionally, sharing or validating insights involves exporting data and communicating through separate channels, adding further complexity and workflow disruption.

With all this cognitive complexity and disruption, it's probably already clear to you why BI adoption among business users remains low (less than 30% of users according to analysts) even while vendors are excited about their success among more specialized users.

AI and BI

It is tempting to think these issues with Business Intelligence might be solved with Artificial Intelligence. Perhaps AI can make the right discoveries in data and even make automated decisions without being distracted by changing contexts?

In practice BI and related data management technologies remain essential for their foundational strengths in data management, reporting, and analytics. An enterprise data model does more than just bring data into one place for analysis. The data engineering process includes cleaning data, conforming it to standards and applying business rules. For example, your financial reporting, your HR data and your customer data all have rules about access and rules about how they are to be correctly interpreted or aggregated.

Much of Business Intelligence focuses on frequently accessed operational reporting of clearly defined OKRs and KPIs which needs high trust. AI generally adds most value in more innovative exploration. So businesses are rightly excited by the creative power and insight of AI and its ability to work with natural language. But nevertheless, they are wary of AI's drawbacks: you don't want your tax reporting to be a hallucination!

With its remarkable natural language capabilities, AI holds out the promise of an engaging and insightful user experience. That is an exciting prospect. But a well-built data model defining your business terms and semantics is also essential, both to provide traditional reports and BI and to serve your AI with reliable data. (See the section Embedded AI and the Semantic Layer below.) The synergy of AI and BI together enables a new approach data-driven decision-making.

We can now see the emergence of that synergy. AI-powered chatbots built over carefully curated business semantics are an excellent way to embed analytics for business users who need to ask new questions or to explore ad-hoc ideas. But traditional dashboards and visualizations remain the most efficient way to present structured data for regular review in a consistent format.

The lesson is that if we want to enable every user to make better decisions with the right information, we must meet them where they are. We need to bring analytics out of specialized workflows and into the operational context of typical users. Whether working with AI, BI or both, we must embed analytics in workflows and applications where the real business happens.

Embedded Analytics and Operational Applications

This is the challenge I set out to address with Jim Horbury in 2023, when we wrote our O'Reilly book, *Embedded Analytics: Integrating Analysis with the Business Workflow,* where we said "We think embedded analytics is an exciting field that promises to transform the way we design, build, and use software applications. It is a field that brings together the worlds of data analytics and application design."

That last sentence is important. Embedded Analytics integrates data analytics and visualization capabilities directly within a user's natural workflow as part of the design of the host application: in effect, within the user's field of vision, without distraction or task-switching. This also brings analytics development within the scope of application developers rather than data specialists: an important shift in practice.

As Embedded Analytics uniquely brings analytics and business actions together, I could list endless potential use cases. In fact, it's difficult for me to see where analytics does not add value.

Some of the most common scenarios are internal analytics – where organizations analyze their own data for their own use. For example, CRM systems utilize analytics for personalized marketing and customer service by understanding customer behaviors and trends. Similarly, supply chain management, HR, finance and e-commerce can benefit greatly from new insights.

However one of the most compelling use cases for embedded analytics is to enable customer-facing analytics or external analytics.

Embedded Analytics Use Cases

I am personally passionate about the value of analytics for business. In fact, I often say:

Data without analytics is a wasted asset, but analytics without action is a wasted effort.

Customer-Facing Analytics

As a consumer, you may already be familiar with some customer-facing analytics. If your utility bills include visualizations of usage over time, or comparisons between your usage and others in your neighbourhood or zipcode, then you're using Embedded Analytics!

Similarly, many software companies now want to deliver additional insight to their customers by embedding time-series analysis, comparisons, predictions and recommendations within their applications. This enables business users to identify areas for improvement, track progress over time, and strategize effectively based on comprehensive, data-driven insights.

In these cases, the software vendor's own developers do the work of defining the key analytics and integrating the user experience. When done well, the result is a user interface in which the analytic components are seamlessly integrated with the host application. For the user, this is simply the same experience.

You can read more about the Cube architecture in this paper, but there's an important point to be made right here, for customer-facing analytics. You don't want to embed a heavy-weight, inflexible analytic engine into your carefully crafted application or service. Yet the footprint of some analytic platforms is just too cumbersome for embedding. Cube is powerful enough to serve demanding analytics, embedded for external users, but with a footprint that will not bloat your own application or service.

The architecture to support each use case often depends on the integration capabilities of the host application, the existing data infrastructure and the skills of your developers. However, there are some important shifts occurring in the industry right now. More applications ship with APIs which enable embedding and data engineering pipelines and data ops increasingly allow application developers to work with data where previously specialized data developers were needed. These changes allow for a better approach: using a Semantic Layer to deliver context-rich business data to libraries of embedded visualization or report objects . We'll look at this architecture in detail later, specifically as implemented by Cube. First let's review the advantages and disadvantages of some other Embedded Analytics architectures which you may come across.

Embedded Analytics Architectures

There are four broad approaches to embedding analytics:

- Hosted reports
- Embeddable BI platforms
- Embeddable analytic libraries
- AI-powered chatbots

1. Hosted Reports

Even a small business using a simple accounting package expects to have reports of some kind available in the application. As consumers, we have similar reports along with our credit card statements or utility bills: the principal is the same. These reports may be hosted on a separate page or tab of the application. Today they are increasingly graphical and have simple capabilities such as sorting and filtering: not quite analytic, but not merely static printouts either.

In the enterprise, a key benefit of embedding a reporting platform for Embedded Analytics is simply consistency across the organization. For users who work with business reports in other contexts (managers, for example) the embedded versions will appear familiar and easy to navigate.

Nevertheless, reports offer very limited insight: convenient and easy for sure, but that's about it.

2. Business Intelligence Applications

There are now many BI applications available on the market ranging from simple visualization tools to complete analytics packages with statistics, predictive analytics and AI augmentation. Many of these tools have some facility for embedded, with some important differences depending on their architecture.

In general, the key components of an embedded solution are:

- The embedded presentation layer
- The analytics engine
- The data sources

Each of these can be developed and deployed independently, but in some approaches your architectural choices will appear limited. Especially in the most sophisticated packages, the analytics engine may be tightly coupled with the presentation layer as a legacy of an older architecture.

There are some advantages to embedding BI applications, especially for organizations with IT-provisioned BI already widely deployed: familiarity and ease of use due to existing user proficiency with the BI system; dynamic and interactive analytics capabilities; and consistent data reporting across the organization. Centralized management might also simplify governance and compliance. This approach may also look cost-effective in theory, leveraging existing investments rather than requiring new development, but in practice the cost advantage is limited because the embedded solution will involve both new work and new deployment. (Licensing is rarely the most significant cost in any system.)

Even with this approach there are numerous options for embedding.

The key methods include:

IFrame Embedding

This is one of the simplest ways to integrate analytics into an application. It involves embedding an analytics dashboard or report within an inline frame (IFrame) on a webpage or application. Developers find this method relatively easy to implement as it doesn't require extensive coding or deep integration with the application's infrastructure. But the main limitation is that IFrames remain somewhat inflexible. They often only provide a "window" into a separate system, which does not provide the best user experience or good interactivity with the host application.

White-Labeling

This approach involves rebranding an existing analytics solution while shipping it with the host application. The embedding may use one of the techniques described above, or the applications may just sit side-by-side. White labeling offers a way to provide analytics services under the host application's branding. It's also useful for software vendors wanting to offer analytics as part of their own product suite.

API-Based Integration

Another approach is to use APIs to integrate analytics directly into the application. The analytics platform exposes APIs to allow the operational application to request and receive data and visualizations. In particular you may develop a look-and-feel which is more integrated with the workflow of the host application. But of course, this requires technical expertise and development effort. The integration needs to be maintained as both the analytics tool and the operational application evolve. It's also worth noting that not all BI platforms have good APIs. Some of your favorite features in the platform may not be well supported, if at all.

Embedded BI Platforms

Some BI platforms have been designed from the ground up to be integrated into other applications. These platforms provide comprehensive analytics capabilities, including data visualization, reporting, and dashboards which can be embedded with sophisticated tools for integrating both workflow and look-and-feel.

3. Embeddable Analytic Libraries

Pre-built, reusable components can provide developers with an efficient way to build analytic features into operational applications without the need to code common features from scratch. Analytics libraries may include data visualizations, predefined metrics, predictive models, interactive tables, report formats and many other useful objects. These libraries facilitate rapid development and deployment, thanks to their pre-built, ready-to-integrate components.

Most importantly, component libraries enable the most complete integration of operational and analytic workflows.

However, unlike a complete BI platform, component libraries may not have their own analytic or semantic engine. This may sound like a significant disadvantage, but in practice it affords some new opportunities.

Remember first that the tight coupling between the analytic engine and the presentation layer is exactly what makes embedding BI platforms so frustrating for developers. Also, the analytics engine of a classic BI platform may exceed the simpler requirements of an embedded scenario.

Fortunately there is another approach. A headless analytic engine or Semantic Layer decouples the presentation layer from the analytic logic and data engineering. We'll talk about that in some detail later.

4. AI-Powered Chatbots

Perhaps the most modern and user-friendly way to embed insights in an application is through a chatbot. These friendly interfaces used to be limited in functionality because of the difficulty in translating from natural language to data, finding the relevant answers in a knowledge base, and then delivering the detail back in natural language again.

However, with the availability of ChatGPT in 2023, the experience of chatbots has changed forever. Now we can reasonably expect to interact with natural language and get back genuinely relevant insights which may be in natural language, or with multi-modal AI, may be new datasets or visualizations.

The challenge for developers embedding an AI chatbot is not the user experience, but rather ensuring that the Large Language Model (LLM) driving the bot has access to the relevant data and context for the business. But developers also need to constrain the LLM to give only the relevant answers and detail. Prompt-engineering can help, but it's not enough. Cube's Semantic Layer provides the LLM with a model of a world, composed of entities and their measures and dimensions. It's a powerful way of enabling a chatbot experience that is both usable and relevant.

Clearly, the Semantic Layer is critical to the success of many of these embedded scenarios, so it's time to look at this feature in more detail.

The Semantic Layer in Embedded Analytics

The Semantic Layer is a critical middleware sitting between the data source and presentation layers in an analytics stack. It manages and processes the information fed into downstream data applications, ensuring efficient building of native applications and data consistency across different platforms.

The Semantic Layer addresses several key challenges:

Data Consistency and Control

It centralizes data modeling, metrics, and access controls, ensuring data is consistent across various applications and users. For instance, Net Sales in a dashboard will mean the same in a mobile CRM application, enabling different users to work with the same data in very different contexts.

Performance and Cost-Efficiency

By pre-aggregating and storing data, the Semantic Layer can address slow application performance. Advanced caching can also enable the kind of high-concurrency, low-latency queries we're seeing so much demand for now. This can also bring down expensive, high-end data warehousing costs.

Streamlined Development

Some BI platforms have been designed from the ground up to be integrated into other applications. These platforms provide comprehensive analytics capabilities, including data visualization, reporting, and dashboards which can be embedded with sophisticated tools for integrating both workflow and look-and-feel.

Flexibility and Integration

A Semantic Layer should seamlessly integrate with tools and sources through APIs, supporting a diverse range of experiences and data applications.

Data Governance and Security

The Semantic Layer can be instrumental in enforcing data governance policies and security measures. By centralizing data access controls, without requiring IT or users to configure every back-end system independently, the Semantic Layer ensures sensitive information is only accessible to authorized users, thereby maintaining data integrity and compliance with ever-more-demanding data privacy regulations.

Aside from embedded scenarios, the Semantic Layer also has advantages for conventional BI users, readily integrating new data sources, reducing dependencies on IT for analytics and translating complex schemas into business terms and semantics. When embedding analytics, the application developer will typically take care of those issues, but they remain useful for wider adoption.

Not All Semantic Layers Are Equal

I have written about Semantic Layers in general, but in reality, there are quite a variety of options.

Some Semantic Layers are simply architectural components within an existing BI platform, exposing cleaned and structured data for the BI developer, and they are therefore limited in scope for customization as well as being locked-in to the vendor.

Some Data Warehousing platforms also include Semantic Layers as a way of sharing metrics and business definitions. However, in a Data Warehouse, a Semantic Layer is rarely a specifically engineered component and more often just a way of developing, collating and managing views over existing data structures. As a result, performance and scale can be problematic, while maintenance and development demand extra effort.

An independent Semantic Layer is much more promising for our purposes, giving application developers the tools they need for defining reusable business semantics with high performance and scale, but remaining decoupled in logic from both the back-end data sources and the front-end applications.

Of these independent Semantic Layers, Cube has some specific advantages emerging from its unique features and design principles, which are tailored to meet the challenges of modern data analytics, but also from a deep understanding of application development.

Cube for Embedded Analytics

Cube, as the team describe it themselves, acts "like a centralized control panel for your company's data." That is to say, it manages data modeling for defining consistent semantics, access control for security and governance, caching for performance and scale and APIs for developer integration.

Delivering consistent business data and enabling developer productivity are considerable advantages in themselves. But the landscape of Embedded Analytics still presents challenges.

Data Integration

With the increasing variety and volume of data sources, integrating the range of data needed for decision-making into a cohesive framework is still a complex task. Traditional BI systems struggle with this integration, leading too often to data silos and inconsistencies as ad-hoc solutions spring up across the enterprise.

Scalability and Performance

It's not only the complexity, but the sheer volume of data now being generated, that can cause problems. BI tools can become their own bottleneck.

Real-Time Data Analysis

Increasingly, business users expect insight to data in real time. This is particularly true in Embedded Analytics, where the users quite reasonably want the analytic experience to be completely in sync with the operational applications in which it is embedded. Given the demands of data integration and scalability, near-real-time analysis is an important issue to tackle.

User Accessibility and Usability

As data analytics becomes integral to more roles within an organization, tools must cater to a broader range of technical expertise. Traditional tools often have steep learning curves and are not user-friendly for non-technical users. In Embedded Analytics, the learning curve needs to be appropriate for the skills of the operational user. You can't expect a call-center operator, for example, to take a specialist analytics training course.

Cube does have some unique approaches to these problems that are both innovative and practical. In particular, Cube's API-first architecture allows seamless integration with various frontend and backend technologies. By providing APIs like SQL, GraphQL, and REST, Cube enables different applications and tools to access and manipulate data efficiently even from highly disparate data sources. This approach also enables businesses to embed analytics with custom applications and to use their preferred tools for data visualization and BI while relying on Cube for data management and consistency.

Cache Talks

For scalability and real-time analytics, Cube's advanced caching not only improves query performance, but is particularly effective in environments with high-concurrency queries: multiple users performing data operations simultaneously. This is a very common cause of problems in Embedded Analytics where there may be many instances of the business application in use at any time. All too often, with regular BI tools, this not only results in poor performance for the analysis, but also puts extra load on the operational systems which run the business.

By caching query results effectively, Cube reduces the number of direct queries made to the underlying data sources. Cube also provides customization options for its caching mechanism. Administrators can configure cache duration, size, and other parameters based on their specific requirements and usage patterns. This level of control allows for an optimized balance between performance and resource utilization.

Embedded AI and the Semantic Layer

I earlier said that AI and BI are complementary technologies for business. I think it's also clear that AI is useful as a natural language interface for BI. We also see many BI vendors implementing what they call Augmented Intelligence, where machine learning finds patterns or makes predictions based on BI data. In short, we see in practice AI for exploration and BI for trusted reporting.

However, in order for the AI to truly add value in understanding BI data, it needs enough context and semantics about that data to avoid inaccuracies or hallucinations. This is why semantic layers are so important – they provide the definitions, metrics, entities, relations and context for LLMs to properly understand the data on which they must base decisions or answers.

By structuring and organizing business data into meaningful terms rather than just database schema, the semantic layer gives LLMs what they need to accurately interpret BI data. For example, Cube and Delphi Labs working together have shown in benchmarking that LLMs connected to semantic layers can improve answer quality and accuracy from 16% to over 80% when compared to a naive Text-to-SQL approach. So while AI promises to take BI analytics to the next level, semantic layers are the crucial element enabling AI to leverage BI data properly.

This increased accuracy may be useful – but not critical – for exploratory analysis such as modeling new marketing strategies. But of course it is very relevant to Embedded Analytics where KPIs, trends and predictions are being used in an operational context. A customer service representative on a call might query the system for the lifetime value of a customer and an analysis of their current spending patterns: they need an accurate answer, not a hallucination.

There's another twist to this. Embedded Business Intelligence is often implemented by developers with a specific background in data engineering or visualization. AI experiences are increasingly built by application developers who may not do their own data engineering. So the Semantic Layer implementation really needs to be accessible to a new community of developers.

With this in mind, let's look at how you might get started with Cube's Semantic Layer, for Embedded Analytics or AI...

AI experiences are increasingly the domain of application developers who may not do their own data engineering. So the Semantic Layer implementation really needs to be accessible to a new community of developers.

Getting Started

Cube Cloud greatly simplifies the process of building, testing, deploying, and managing Cube projects. It comes with a free tier suitable for development and proof-of-concept project. Additionally, Cube Cloud provides a range of managed infrastructure and workspace tools designed to enhance team collaboration and project efficiency, such as a web-based data model editor, BI tool synchronization, auto-scaling, and observability features. It's a great way to get started.

Cube's documentation includes 4 excellent guides to getting started.

- For Snowflake users
  https://cube.dev/docs/product/getting-started/cloud
- For Databricks users
  https://cube.dev/docs/product/getting-started/databricks
- With a demo deployment
  https://cube.dev/docs/product/deployment/cloud/deployments#demo-deployments
- With a completely fresh deployment of your own
  https://cube.dev/docs/product/deployment/cloud/deployments#creating-a-new-deployment

Give it a Try

I would suggest trying the demo deployment first. It simply involves creating a Cube Cloud account connecting to the DuckDB demo source with sample data and a pre-configured data model. This demo is an excellent way to familiarize yourself with Cube's functionalities, including dynamic data modeling.

The platform offers a comprehensive overview of each deployment, detailing API endpoints, resource allocations, and recent activities. It also displays user-specific states like the current source code branch and development mode status. Cube Cloud efficiently manages resource allocation based on activity levels, ensuring optimal usage for different scenarios. In short, unlike a lot of demo environments, you don't just get to play with the data, but you'll get important insights to the administrative and management features too.

What's stopping you?

Conclusion

The world of analytics has come a long since my own first naive attempts to make sense of data. And we are now on the brink of another new age of technology. But two things have not changed: the need for business to have a shared, trustworthy understanding of their own processes and the state of the business; and the need for explorers, innovators and the curious in every department to be able to do their own analytic work and to make new insights.

Importantly, AI has not reduced the need for analytics, but if anything it will increase the demand. Users will reasonably expect applications and platforms to be more insightful in every scenario, but generic AI can not achieve this on its own: it needs your specific business context and rules in order to give precise and relevant insights.

And all this must be embedded in the users' workflow, whether those users are internal to your organization, or external customers.

I believe that Embedded Analytics is a critical approach for enterprises who want to make data the currency of everyday decision making. And, as I hope I have shown, the Semantic Layer, designed for application developers, is exactly the way to make this happen.

No technology is future-proof, but equally no business can afford to sit still these days. Embedded Analytics built over a Semantic Layer promise an excellent way to enable the highest value right now and through the next phases of business transformation. It's going to be a wild ride, but this is a practical path.

TRY CUBE FOR FREE

About Cube

Cube is the semantic layer that makes it easy to break down data silos, create consistent metrics, and deliver them to all your data endpoints – BI tools, customer-facing embedded analytics, LLMs, and AI agents. Cube Cloud delivers the Enterprise-ready semantic layer that connects to any data source. Robust developer tools, observability, security, and compliance make it easy to quickly deploy, monitor, and use Cube across any sized business.

Companies such as Drift, Cloud Academy, Intuit, Walmart, Security Scorecard, and IBM trust Cube to deliver amazing data experiences to its their customers and employees.