

HOW TO CHOOSE THE RIGHT SOFTWARE STACK FOR YOUR DATA PROJECT

INTRODUCTION

Ever heard that stat about humans only using 10% of our brains? [Well, I'm afraid it's a myth.](#) While it's true the 100 billion neurons in our brain are not firing off at the same time, the brain uses a fifth of our body's energy. So imagine trying to run the whole supercomputer brain at full capacity all the time? It would take a serious amount of power.

Comparing brain usage and data usage is a great way to understand data analysis. Over the course of your day, pretty much every corner of your brain will come into play in some capacity. You draw on information from every nook of that vast and complex network just to navigate everyday challenges. Much like your brain, the software you use to connect and analyze your data needs to be able to jump to wherever the data is stored for the task in hand.

Let's take another stat you might have heard: Of all the Big Data accumulated, [only 0.5% gets analyzed.](#) Now, while the data stat is true, it doesn't mean that only 0.5% of the available data is useful to you.

And that's where the problem lies.

Just as seemingly snap decisions are really based on incredibly sophisticated webs of information, the most useful, actionable insights about your business are the result of synthesizing and reconciling sources from all different places. So, how can you start treating your data the way your body treats your brain? It starts with choosing the right software stack for your data project.

Contents

1

What's the Project?

How Complex Is the Data?

Short-term or Scalable?

Who Needs Access?

Business Intelligence vs Dashboard Reporting

2

Comparing Software

Front End, Back End and Joining Everything Together

Data Warehouses

Data Marts

Single-Stack Architecture™ (Elasticube™)

3

Final Thoughts

Next Steps



WHAT'S THE PROJECT?

Before you can decide what software stack you need for your BI Solution, ask yourself the following questions:

How Complex Is the Data?

First up, let's define what kind of data project this is. Is it a relatively straightforward project that uses a handful of easily reconcilable data sources, or a one-off analysis you'll never need to repeat again?

Or, more likely, is this a more advanced project, using multiple, scattered data sources? A long-term solution, perhaps, that you'll keep coming back to and refining as your needs evolve?

Short-term or Scalable?

If you're using predictable, consistent data reports based on single or simple data sources, you might decide that a project-specific approach is best. Basically, this is for projects with limited scope that you won't need to build on or adapt in the future. If you're looking at a long-term project, are dealing with all kinds of different data, or expect your need for reporting to get more complex as data accumulates, it's best to go for a solution-oriented approach.

For the latter, you should be on the lookout for a software stack that's flexible and scalable. It should be able to tackle what you need it to do right now, but also

cope with future reporting requirements as the business grows.

Who Will Tie the Strands Together?

Don't forget that every system you use as a data source is likely to have its own way of storing and presenting information. You're going to have to find ways to join up - and clean up - all that data before you can do anything with it. Imagine that every scrap of information you're importing is like a book that exists in a ton of different editions. If you were trying to figure out how many copies of this book are on sale right now, you'd need to figure out how every single different bookshop, library, online store and second-hand stall tracks its inventory. You'd need to find out if they list by the author's name, by the title, or by ISBN. You'd need a way to tell when the same book is showing up as two different books, perhaps because of the way the information has been formatted, or its publication date, or who wrote the foreword or whatever.

You'd have to decide whether it counts if the book is included in an anthology and if so, how to factor that into your records. And then, of course, you'd need to figure out how to iron out all these inconsistencies when the information is pulled into your own system, so that you can tally up the total number of books at the end.

The problem isn't finding tools that

can do what you need. Whatever your requirements, there will be the tools and software out there to make it happen. The problem is building them into a single workflow.

Taking a mix and match approach to tools that handle different parts of the process (preparing data, managing data, visualizing data and so on) is fine, but you need to have a watertight strategy for getting them to work together seamlessly.

The issue isn't just which of these functions you need, but whether you have the capacity to blend them yourself, in-house. If, realistically, you don't have the resources, skills or time to do this well, you'll need to source a vendor who can handle it for you.

Who Needs Access?

Next up: who needs to use the system you're setting up, day-to-day? And what kinds of business questions will they be asking?

Unless your IT team will handle all queries and analyses (not exactly the most efficient use of anyone's time and a high risk for a bottleneck) you'll need a self-service system in place that's powerful and user-friendly enough for non-techies to handle independently.

And, of course, you need a system that can handle these questions at speed. The software stack you use must be able

to process the volume of data and the complexity of the tasks at hand without grinding to a halt – otherwise your team won't use it, no matter how useful the insights.

Business Intelligence or Dashboard Reporting?

Some businesses that need to make data analysis available to their non-techie staff try to get around the problem by sticking to straightforward dashboard reporting. These dashboard or report projects are kind of limited. Typically, they're used to address current requirements rather than future ones. They tend to be static - you create them once and then update them with fresh data further down the line, but you can't change the actual functions or types of calculations they're setup to perform.

Business Intelligence is much more sophisticated. You use dashboards to visualize results, but this is just one component, and is nowhere near as restrictive. The business intelligence process includes a ton of work and tools that go into preparing and querying the data, giving you far more scope to ask the questions you want, and to create new dashboards and reports at just the click of a button.

2

THE COGS IN THE MACHINE

So far, we've covered the broad approaches you could take to your project. Next, let's look at the specific software options that go into building your stack.

Front End, Back End & Joining Everything Together

I don't have to tell you how fast data analysis, and the technology that supports it, is evolving.

Until recently, much of the debate was centered around whether you should opt for a back-end software stack, a front-end software stack or a 'full stack', which would deliver both backend and front-end functionality.

To recap, back-end software stacks are the collection of tools you would use to store, transform and manage data – in other words, your back-end functionality. This is where you run your ETL (Extract-Transform-Load) capabilities, meaning that you pull together data from all your data sources into a single, central database. Elements of your back-end stack might include data warehouses or data marts, which we'll talk about more in a moment.

Front-end software stacks are the elements that the end user interacts with, like data visualization, visual analysis and dashboards. The most advanced version of this – the system we offer at [Sisense](#) – is what we

call our [Single-Stack™ Architecture](#) (single-stack software). Instead of taking a back-end stack and a front-end stack and then just tacking them together, a Single-Stack Architecture™ treats the whole process as a single platform from the outset, making sure that each cog in the machine is setup to work together in the most streamlined way possible.

But before we get into that, let's jump back a moment and take a look at some elements that typically feed into a back-end software stack.

Data Warehouses and Data Marts

First up, there are data warehouses (DWs). These are "Big Scale" technologies that bring all your data from different sources together into one place, so that you can treat it as a "single version of truth". The benefit is that the data stored here has been cleaned up for analysis, is up-to-date and ready to use.

Of course, the trouble with using huge data sets is that they take a very long time to load and process. Different types of DW technology get around this in different ways:

- Columnar Databases allow users to pick out exactly which pieces of data they need and only load that data. Meaning, you don't have to wait for your computer to try and process everything in the warehouse to get the information you're after.

- Distributed Databases, on the other hand, work by spreading the load across many different computers, reducing the pressure on each one and make it far easier to scale out.

Data Marts

Another way to get around the problem of handling enormous data sets is to grant access to specific parts of a data warehouse to different teams via a “Small Scale” technology called data marts (DMs). These could link to your own DW, or you could save money by opting to rely solely on DMs that link to existing DWs without owning an in-house one yourself.

While DMs handy, they come with obvious limitations: By cutting them off from the full scope of enterprise-wide data, you cap the kinds of questions and types of analysis each team can perform... which defeats the purpose of having all that knowledge in your company in the first place.

Plus, while DMs are supposed to keep things manageable, they’re not great at dealing with complex queries. As your data sets grow and more information pours into the mart, you can soon find that they get too big to handle.

To function, DMs usually make use of In-Memory Databases or Online Analytical Processing (OLAP) cubes.

- In-Memory Databases reduces the pressure that querying puts on your

hardware by storing data in the computer’s (more plentiful) main memory, rather than using up disk space.

- OLAP tools work by introducing a server between the client and a database management system. The OLAP server understands how the data in the database is organized, so it essentially acts as translator between what the user wants out of the data, and the language and set-up of the database itself.

OLAP cubes make running queries super-fast and easy – but are a double-edged sword. Because the aggregated values they offer up are so clean and polished, you can’t view the data or calculations that go into them, limiting their use to analysts. You can get around it by switching over to view the raw data in SQL, but it’s a pretty inefficient way of doing things.

Should You Incorporate Data Warehouses and Data Marts into Your Stack?

DWs and DMs are great, but they have their limitations – especially when it comes to analyzing and presenting information in a way non-IT teams can understand. This means that, if you’re using a data warehouse, you will still need an agile Business Intelligence tool that your business-focused teams can use to generate and visualize the reports,

analyses and dashboards they need.

If you're using a single stack, on the other hand, you can build the functionality of a DW directly into the platform, while combining elements of both Columnar and Distributed databases to keep the whole thing running as quickly and smoothly as possible.

Single-Stack™ Architecture (the Elasticube)

In fact, that's why we at Sisense decided to name our end-to-end BI software Single-Stack™ Architecture, since it provides a complete, single solution for data preparation, analysis and visualization, but does so without forcing the organization to rely on specialized personnel, or maintaining multiple separate components, modules or tools.

Elasticube is the data store at Sisense's center that works by interfacing between traditional back-end and front-end functions. Elasticubes are extremely fast-running data stores that can cope with high volumes of complex queries – the kind of analysis you need to run for business intelligence. In fact, if you use an Elasticube, in most cases you don't need a

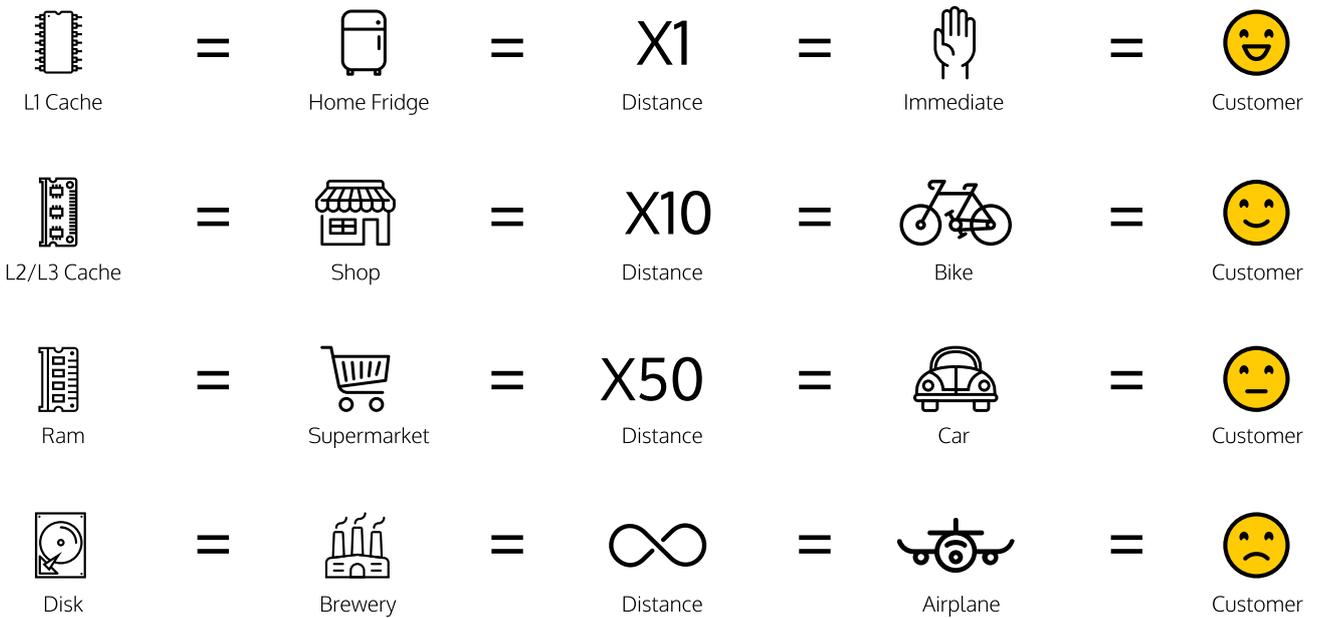
dedicated data warehouse at all.

Like DWs, the Elasticube is designed to pull in data from all different sources, merge the data, and allow users to manually reconcile differences in the way this information is categorized and presented. Basically, you can perform ELT functions and create relationships between data using a visual environment that doesn't require deep-level technical knowledge to navigate or manipulate. You then treat this as a single data set, as you would when dealing with a DW or DM. Here's where it gets really clever, though. When you use an Elasticube, you have access to the full scope of data, but you only actually dip into the data you're using right now.

This is because Elasticubes use Columnar Database technology to ensure that only fields referenced in the query are loaded into memory. The system doesn't need to perform pre-aggregation and pre-calculation tasks that otherwise require all data you might need to be stored on a local hard drive for you to access it. At the same time, Elasticubes borrow from In-Memory Database DM technology, using memory rather than disk space to store data as it's in use.

This cuts down the time it takes to import and process data, reduces the amount of storage space you need, and lets users run queries on pretty much any standard PC that has commodity hardware. It also means you have the benefits of data marts, without the limitations.

IF DATA EQUALS BEER, THEN DATA STORAGE UNITS EQUAL:



3

FINAL THOUGHTS

What's that we hear trumpeting and stamping its giant grey feet in the corner? Ah, that would be the elephant in the room: Cost.

It goes without saying (but we'll say it anyway) that simpler tools are cheaper. A lot cheaper. Opting for a software stack that lets you combine basic dashboard reporting with low-intensity data marts will be a heck of a lot easier on the ol' budget than a sophisticated single stack solution.

And perhaps that's perfectly good enough for your purposes. Perhaps you're just after a quick and easy solution for a simple or one-off project. In which case: don't splash out on what you don't need.

But if your data sources resemble the human brain, with millions of pieces of information scattered all over the place, all contributing to a realistic picture of how your business is doing – you need a software stack that's going to be able to handle that. One that is designed for the people who most need to use it, that can scale with your business, and is flexible enough to take on new data sources as and when you need them.

Yes, it's true that only 0.5% of data currently gets analyzed. But if you want to become the brainiest in the business, you need to be the one to buck the trend.

NEXT STEPS:

Here's your [Free Trial of Sisense](#) BI solution

Looking for a way to compare BI solutions? [Download our BI Comparison whitepaper](#)

START YOUR FREE TRIAL OF SISENSE NOW