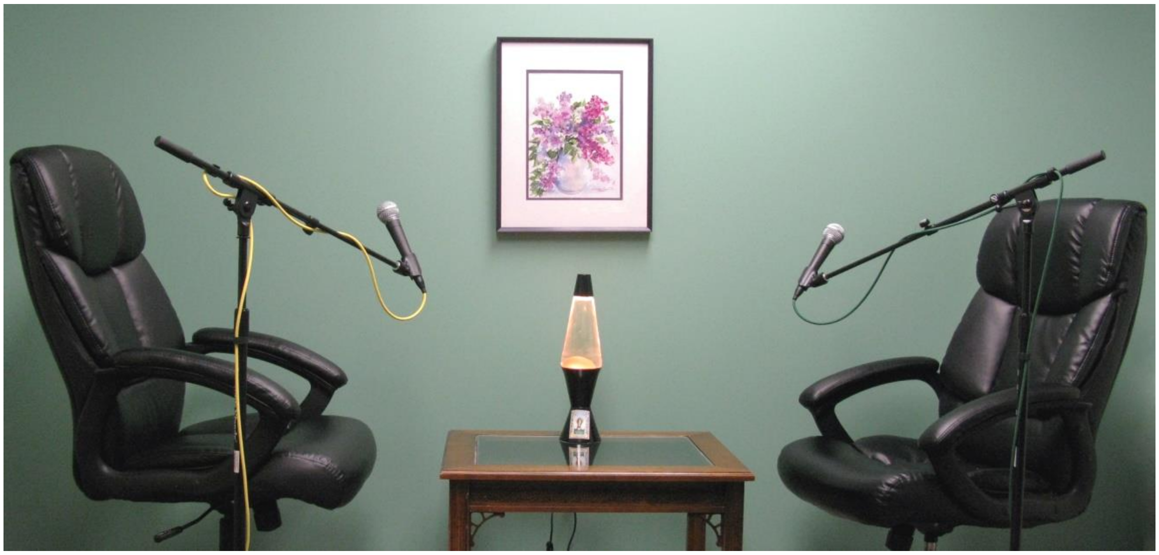
Boulder BI Brain Trust

Leveraging
Business
Intelligence

BBBT Podcast Transcript



About the BBBT

The Boulder Business Intelligence Brain Trust, or BBBT, was founded in 2006 by Claudia Imhoff. Its mission is to leverage business intelligence for industry vendors, for its members, who are independent analysts and experts, and for its subscribers, who are practitioners. To accomplish this mission, the BBBT provides a variety of services, centered around vendor presentations.

For more, see: www.bbbt.us.

Vendor:	**SQLstream**
Date recorded:	**October 4, 2013**
Host:	**Claudia Imhoff**, President, BBBT
Guest(s):	**Damian Black**, CEO
Run time:	**00:15:45**
Audio link:	SQLstream
Transcript:	[See next page]
Transcribed By:	CastingWords

Claudia Imhoff: Hello, and welcome to this edition of the Boulder BI Brain Trust, or the BBBT. We're a gathering of international consultants, analysts, and experts in Business Intelligence, who meet with interesting and innovative BI companies here in beautiful Boulder, Colorado. We not only get briefed on the latest news and releases, but we share our ideas with the vendor on where the BI industry is going and help them with their technological directions and marketing messages. I'm Claudia Imhoff, and the BBBT Podcasts are produced by my company, Intelligent Solutions.

I'm pleased to introduce my guest today. He is Damian Black. Damian is the President and CEO for SQLstream. So, welcome Damian.

Damian Black: Thank you. It's a great pleasure to be here today.

CI: It's been a most interesting BBBT. In fact, I would have to say it's probably one of the most interesting ones. We'll get to that. First off though, tell me a little bit about SQLstream.

DB: Well, SQLstream is an operational intelligence platform. It's a streaming platform -- streaming Big Data platform -- that processes both structured and unstructured data that stream off service data, log file data, sensors, devices. We stream it through continuous queries.

We perform transformations, analytics, aggregation, and let you visualize the information in real time to analyze it, to do all your analytics on the fly, and also to be able to allow your service to respond and react to try and close the loop in real time.

CI: What I liked, you gave us an example, or kind of an analogy of the data warehouse versus this new operational intelligence platform. You said, and I don't disagree with this, I think it was a good analogy, that data warehouses start from now and go backwards. They continue to analyze data into history.

What you said about SQLstream, more of this operational intelligence, is that you start from now and you go forward. You're actually looking at events that are coming into the organization from this period to some period into the future. Would that be a good way of expressing it?

DB: Absolutely. In some ways, we are the mirror image or the opposite of a database. A database is, you run the query now and it looks back into time to its retention policy for how long you have stored and then you come back with your answers.

With SQLstream, when you run the query, the query, again, is from right now, but it's going forward in time. It's querying into the future. The database queries run on demand and they come back with the answers. What we do is we run continuously.

It's absolutely the mirror image. It's running continuously into the future and streaming out incrementally, continuously, the new answers that correspond to the incrementally newly arriving data.

It's not there to replace databases, it's quite the reverse. It's there to make them better, to enhance them. It's to extend the reach of SQL out into the network and to extend the time coverage into the, not just the past, but always looking into the future as well.

CI: Yeah. It's a different technology. It doesn't replace the data warehouse because the data warehouse has its own beauty and its own need. It is a unique technology that does allow us to analyze real-time data in a real-time fashion.

I think that's the part that is fascinating to me, and also something that is completely outside of the data warehouse environment. It doesn't belong there. It is in its own world.

DB: Yeah, I think it's, again, to make data warehouses better, to make them more responsive. People have bolted on ETL and other non-declarative technologies onto this beautiful declarative data warehouse technology.

What they really needed was a beautiful declarative streaming technology, which is the perfect complement to keep them updated, and to do all of the continuous data ingestion, validation, and aggregation, and to give those real-time streaming analytics that are there to complement the historical analysis that you wanted.

If you think about applications like fraud, one that everyone can think about, you have to look for those instances of fraud. Those queries have to be running continuously. It's much better to have those running continuously, looking for the patents that you want to find rather than repeatedly polling all of the past data.

When you find the potential fraudster, you want to then see, "What have they done in the past?" The same thing is true for cross-setting and promotions. You want to look for all of the characteristics of a buyer of my product or service, and, "How can I detect that I have someone that might be a good prospect?"

You'll want to complete it, when you get that prospect, that particular person fits your profile, you'll then want to find out, "What have they bought in the past?"

If you look at all of your applications, you'll see that there's this yin and yang, where there is a continuous piece that you want to do and then there's an historical piece. You want to bring those pieces together to get the full picture.

CI: Let's talk briefly about the three pain points for the need. In other words, there is a need for streaming, especially streaming Big Data if, we have to bring that term in. What do you see as these pain points? Why did this technology come about?

DB: Things come in threes. We like to think of it that way. The first one is just the exponential growth in volume. Not only have we got a problem of exponential data growth, but when you're trying to do it in real-time, it becomes even more acute. It forces you to do parallel processing.

How do you stay on top of this, not only the current volumes, but they're getting bigger every day. You need to have something which is highly scalable. Matter of fact, this is where transaction-based approaches do not work, even Hadoop is showing that. It doesn't transact to get its scaling.

You need to have stuff which basically preserves the original data state and just builds upon it. That's what streaming is all about.

Hadoop is essentially a platform for processing stored streams, which you append to. We're a platform that processes the incoming live streams.

The second pain point is one of business agility, when you're forced then to build this complicated infrastructure that's got to process ever-growing volumes of data in real time, how on earth can you respond fast enough to new market opportunities or competitor threats?

It's going to take you months, maybe even years to build technology that works in scales to respond to the analysis, analytics, whether it's building a real time e-commerce promotion systems, whether it's detecting fraud, whether it's delivering promotions and advertising, whatever it might be. It's going to take you a long time to respond using your existing conventional technologies.

Whereas, what we believe you can do, you can go as fast as you dare move now. If you still want to test to make sure that the results are correct, you can do things in hours and days that would otherwise have taken months and years.

The last point is just the difficulty, the complexity of it. Ironically, hardware is cheaper than ever before. You get multi-core smart phones. You have lots of inexpensive memory and high performance networks. You have parallelism and the ability to process data like never before.

If you try and use Java, multi-threading, C++, program active, procedural logic, you need to get the fastest guns in the west. You've got to pay a lot of money. Even then, you may fail. It's very complicated. It took us years to refine the core of our system. Its parallel processing, scheduling engine. This is very difficult stuff.

What we allow you to do is to define high level, familiar, SQL queries and let the system extract the parallelism automatically for you, perform optimization, and guarantee the answers are going to be preserved, and correctness will be preserved. Take all of that pain off your shoulders so that it will take advantage of this inexpensive hardware and deliver what you need in terms of real time analytics.

CI: Let's turn to the more difficult, I guess, technical concepts behind what you're talking about. A lot of people have heard the term, "complex event processing." They think they understand it. They may not.

Other people have heard the term "event stream processing." They may think they understand it and maybe they don't. I think there is a real distinction between these two forms of real time analysis. I think that's something that people need to understand.

We've got about three or four minutes left. I'd kind of like to dive into that in a little more detail. First of all, why don't we start with just a basic definition of what is event stream processing versus complex event processing. Then, we'll move into some use cases and the technologies behind them.

DB: I think there is a lot of misunderstanding here. There is a very clear distinction between the two, fortunately. Let's put this on the record, so people can start understanding.

Complex event processing, well it's hard to define it as a market. It's obviously complex problems to be solved, but in terms of technology, what it's been about, is about reacting to changes of state. Then, having rules that get fired based on those changes of state.

A lot of the origins came right back into, started off many years ago in artificial intelligence and in monitoring technologies of that kind. The bit that's confusing is they're both about changes. Things changing over time.

There's a fundamental difference in the approach that's taken. One's declarative, which is the event stream processing as a declarative approach. It's processing streams of data and generating new derived streams without transacting or side effecting the incoming streams.

That leads to much, massive scalability. That's the approach that allows Hadoop to scale, by the way, it doesn't transact data because whenever you transact data, you have to synchronize data cross systems and you run into problems of speed of light.

CI: To translate, you're talking about changing it in some way?

DB: Changing the state. Changing the value of a particular thing from one value or one state to another. You can look at, for example, a device that's changing in temperature. You can view it as a stream of metrics that are coming out or you can view it as, it was at this time, it was this value. This other time, it was at another higher temperature.

The complex event processing systems process windows, when they're processing windows of time, if they process windows of time, at all. They process them as structures, which then can be queried. It can be transacted. There can be side affected.

While they can solve similar problems, because it's not a declarative approach, because it has this side effecting thing, it has a very much limited, limited in terms of the scalability that can be achieved.

It's not a Big Data technology. It's certainly not a streaming Big Data technology. It's not amenable to automatic optimization. Some of the other good things which is why SQL and relational databases have made it to the fore in Big Data and why SQL has come back again. It's because optimization and extreme scalability are things that people really do want.

CI: Well, let's talk about use cases. For example, when would I use ESP? What's good use case for CEP? You mentioned fraud, that sounds like a really good one for ESP.

DB: Again, I think, if you want to deal with large scale, streaming Big Data, you've got a lot of information you want to process. You want to process it cost effectively and scalably across multiple servers, using multiple cores, then that's when you want to look at ESP technologies, event streaming technology.

I would argue that you want to do the actual processing, as well, with declarative technology. SQLstream allows you do that with declarative standards-based SQL but you can also do event stream processing using Java and other technologies. Actually, within SQLstream, we do support Java, is also a first class language.

With complex event processing, it's going to be more at the lower level scale but it's going to be probably oriented around a particular patterns that your CEP engines will actually be able to support natively.

I can see customization of these for things like financial trading and so on. The important element here is to, if you want to deal with trying to build your own applications that are going to be in the mainstream, that are going to be doing thints, for example in the area of ETL before now. Just extracting the data and turning raw data into information value and validating your data stream, aggregating them, keeping your data warehouse updated.

Then, getting some continuous reporting and analysis. It doesn't have to be a complex application for you to want to get the value. It doesn't have to be complex for you to process extreme volumes. Those are great events stream processing applications. I believe, going forward, it's going to be an extension of the whole data warehousing and database technology. It's going to be viewed as mainstream data management.

The reason is that it's applicable, it's amenable to SQL processing. A lot of the techniques that have been proven over the decades can be applied to this. It can also solve many of the problems, in the past, that have kept tools that have only been able to solve. I think the complex event processing is going to become more specialize into specific state transitioning devices and so on, where you want to look for certain patterns and take specific behavior.

CI: They have a niche, in other words. They have a specific niche that they lean toward.

DB: They have a niche. I think it's a small niche. I think the way that could add and increasingly become valuable is to specialize to very specific industry problems. I think event stream processing will become extremely valuable by becoming a broad horizontal standards-based approached that everyone can adopt and can use to extend and improve the stuff that they're already using today in data warehousing and business intelligence.

CI: Nice complement. Complementary.

DB: Absolutely complementary.

CI: Well, on that, unfortunately, we're about out of time. So that's it for this edition of the BBBT podcast. Again, I'm Claudia Imhoff and it's been a great pleasure to speak with Damian Black of SQLstream today. Thanks so much, Damian.

DB: Thank you very much.

CI: I hope you enjoyed today's podcast. You'll find more podcasts from other vendors at our web site, www.boulderbibraintrust.org. If you want to learn more about today's session, please search for our hash tag on Twitter. That's #BBBT. And please join me again for another interview. Good bye, and good business!