



EXECUTIVE BIO

INNOVATING@SUN

Project Hydrazine

Announcer: You're listening to the Sun Microsystems Podcast Network. Welcome to another edition of Innovating@Sun with your host, Hal Stern. Today's topic: Project Hydrazine. And now, here's Hal Stern.

Hal Stern:

Hello and welcome to another edition of Innovating@Sun. I'm your host Hal Stern, Vice President of Global Systems Engineering, and I'm joined today by resident rocket scientist, Vice President, Distinguished Engineer and rather tall guy, Bob Brewin. Also known to Muppet Show lovers as Dr. Bob. So, Bob, welcome to the show.

Bob Brewin:

Hey. Bunsen Honeydew, how you doing?

Hal:

Okay. And we spoke around last year's JavaOne about the project you introduced there, Java FX. And in keeping with your great software tradition, this year you slipped in another mention at JavaOne in your keynote about something called Hydrazine, and I think a lot of people picked up on it and saw a glimpse of where you see service creation and service delivery going in the future. So tell me a little bit more about Project Hydrazine and what this has to do with rocket science.

Bob:

Well, the rocket scientists – you know, I'll go backward here. Obviously it's interesting is one of the things that Eric Klein is the Vice President of Marketing for the CSG group and I were sitting there in my office chatting about this thing we needed which, you know, we obviously have this fine technology in the form of Java FX which is the idea of creating sort of blended experience and clients across a wide variety of devices. But if that's the rocket, if that's what we're launching, we need a rocket fuel behind it. And Eric, in his lovely fashion, immediately twigged to that and said, that's it. We need rocket fuel. And, you know, e-rocket fuel equals hydrazine, and that sort of begat the name. So the thing about Hydrazine, Hydrazine really is about the back-end of services. In other words, the Sun's developer and consumer-facing platform as a service offering for things like development, collaboration between developers, content offer, the delivery, and through things like Project Insight, the commercialization of those blended consumer services are all those clients talking to the big cloud.

Hal:

Well, why don't we back up. I would say when you talk about clouds, we tend to go back to our drawing in the early days with the Internet boom where we'd draw TCPIP cloud as a – and that's a bunch of ovals on the whiteboard somewhere, and the implication was you had a well-defined set of interfaces to it, but you didn't really need to know what was happening inside the cloud. The packets went in on one side and they pretty much came out in order or at least could be assembled in order on the other side. And what wires or wireless or fibers or routers or other pieces of equipment you went through along the way, blatancy, reliability, geography were sort of left up to the actual providers. It was a cloud. It just kind of worked. And there was assumption there that it worked with a reasonable level of assurance. What's that mean when we talk now about this service delivery to the rich client? What's the cloud look like now?

Bob:

I think in many ways, the cloud hasn't changed really. I mean, the idea being is that I think the most important thing when we start thinking about Rich Internet Applications in a diverse set of clients is the

most important thing to those clients is – there's two things. Obviously the experience, hence, RIAs, but it's the data, right? Where does the data live? That lives on the cloud. It's a phrase we've tend to throw out and not really pay attention to because we're so familiar with it; we see it everywhere we walk around, whether we're in Menlo or Burlington or anywhere, which is "The Network is the Computer". The network really is the cloud, right? So that is the place where that data resides on servers somewhere on the Internet, right? And those applications which are running on your devices, whether it's your browser or your phone, heck, your in-dash DVD system, they're all connecting to the network. They're basically looking to the cloud where essentially their presence, their data, all live, regardless of which devices are connected to it. So I don't think it's really changed that much. I think what's changed is the fact that we actually have different and diverse types of clients. We have a higher expectation in terms of client experience and consumer experience on leveraging those services as they exist in the cloud rather than just using it as a, for the lack of a better term, an uber-router to essentially just send packets through, between devices.

Hal:

And I think that part of that speaks to the economics that if you have a well-defined set of interfaces that you want to go write to, you don't really care how they're implemented. You just consume them. And we've seen this happen in all sorts of things that we talk about being commoditized, whether it's being able to buy commodity gas because your gas tank has the same size nozzle on it no matter who you buy your car from to being able to get electricity out of the wall because at least on a country-by-country basis, you know what the plugs look like. So how much does Hydrazine as a project start to shape what we think those plugs are, what we think the sockets are that we're going to go plug into?

Bob:

Well, one of the things that's sort of at the foundation of what we're trying to build with Hydrazine is this notion that underneath the covers, i.e., the implementation of it, and again, those plugs, those API's who get exposed, are a set of either standard web services or you might say very simple web services, i.e., REST, but the foundation, i.e., the application platforms upon which those services are then delivered, are built on a set of open platforms that we have promoted and standardized over time, i.e., Java, GlassFish, things like OpenDS, maybe OpenSSO, a number of things where clients who were actually going to go ahead and build and deploy services on that cloud, i.e., that infrastructure, are comfortable knowing that they have the flexibility, aren't getting locked into a particular vendor. They know that via Hydrazine and via things like Insight, they actually have a control of the communications and the dialogue they're having with their customers rather than perhaps getting locked into somebody else's business model and somebody else's set of private APIs which essentially keeps them locked into their platform, their cloud.

Hal:

So if I wanted to ask you to pop down another level here and extract some more detail, what are the actual components you're looking at? What are the things that we're going to go deliver here? You talked about building on top of a set of RESTful services as well as building on top of some existing services, many of which I would say predate our notion of REST of HTTP as the primary means for exchanging data with a client-server pair. What's your starting point?

Bob:

Well, there's many levels to what we're building out there and, in fact, Hydrazine, I think, is just one platform as a service offering that Sun will deliver over time. We have others. For instance, people have talked about things like Caroline. People have talked about Darkstar. Within this space, there are things like Marketplace in Sun which is, again, tied in to a lot of what we're doing in sort of the consumer-facing applications coming out. So there's, again, many levels to this thing. I think at the lowest level, we're talking about leveraging our strength in things like OpenSolaris, xVM in terms of building a sort of virtualizing container environment, i.e., an infrastructure as a service. A lot of that will be focused and delivered by Dave Douglas and his Network.com team. Sitting above that, we have a set of constructs that are largely built around open and standard product offerings that we have such as application servers in the form of GlassFish directory, in terms of OpenDS, Database, MySQL, identity servers and things like OpenSSO. Sitting on top of that will be a set of frameworks and platforms that'll allow it to do things like

developer and offer services, right? So things that will allow developers and content authors to go ahead and leverage things like NetBeans to instantly deploy to this running infrastructure, and then deploy the services for others to use. And then over time, I see our sales as well as other companies and partners and any developers that are coming to Network.com start to build things and deliver services like media transcoding services, identity services, database backup services, communication services like email, IM, calendar. Of course, there's no reason why we can't take a lot of that infrastructure and throw that on there as well. Obviously, a big one will be storage services, and I think that's something that the Network.com folks will be paying close attention to in how they can deliver an S3-like experience for people coming to Network.com as well.

Hal:

So you made an interesting bridge there to looking at the state of the state. Clearly Google, Amazon, Microsoft Live, Salesforce.com, we're seeing a number of players now who give you a particular set of APIs and implicit in there is if you just go code to these APIs, look at all the great service functionality we can go provide to you. And I get to throw you at least one curve ball. Being a coast away, I'm pretty sure you're not going to hit it back at my head. What are we doing here that's different in terms of either broadening that set of APIs or looking at where the industry's going around the set of APIs that define what you call "the services on top of the platform"? Where do you think the key aspects or the key attributes are that we want our developers thinking about in terms of how to go create these services?

Bob:

Well, I think there's a couple of them. One of them essentially is the set of APIs and services that we're going to be building on and providing – Sun is, of course, always committed to open standards and lowering that barrier to exit, as Scott had said, about this notion that I can build it there but then, if I want to, I can take my services; I can host them myself, right? I can put them on the cloud. I can take them off the cloud at my leisure. And one of the interesting things around technologies or the Project Insight which is sort of tangentially related is this notion that what I want to do is I want to enable the instrumentation of, for instance, the application and services by developers for their business use, right? Unlike, let's say Google, which is you're basically allowing a company like Google to essentially dis-intermediate yourself by putting them in your business pipeline, right? Which is something different, and again, this goes back to one of our core values at Sun which is making sure that we provide open and opportunistic platform and playing field for developers to come and, thus, building an affinity to our software, to our hardware, to our services, to a lot of the value that Sun provides.

Hal:

Great. So now I want to go back to your comment about essentially offering infrastructure as a service at a low level. So you think about what one of our customers would want to go through, the thought process you want to go through when you look at either cloud computing or moving toward this more abstract mechanism for deploying applications. There's, as you mentioned, the infrastructure, so how do I just provision and virtualize and take advantage of the physical assets I have? There's the core platform, if you will, the things that people program to. There is the service deployment, and then I want to tie this all the way back to the last conversation you and I had on this show about so what's really happening on the device? And how much do you see this being modulated by the future of the Rich Internet Application vis a vis, hey, everything's just a browser and it's all going to be done in REST because that's the simplest way to go move state back and forth?

Bob:

Oh, so you sort of asked four questions in one. That's more of a curve ball than the prior one.

Hal:

Hey, well, consider it an entire "at bat" worth of pitches.

Bob:

[Laughter] Well, you know that I actually do umpire, right? So I'm going to call you on three strikes here. Why don't I answer two questions. Number one is on the infrastructure side, one of the interesting data

points I think we need to highlight is by our offering an infrastructure for service provider, the really interesting opportunity is through the technology onramp to our systems, which means that because we can actually deploy the infrastructure as a service and build a virtualized environment that maybe looks like a Niagara2 or looks like a Rock or looks like one of our more recent [inaudible] generation systems, I can actually then start building and deploying applications as if I had those systems sitting in my data center, and that provides an opportunity for us to go ahead and then sell the systems to our customers because now they can actually prototype test, performance tune and build the systems on the network and then move it again. Again, this notion of transparency. Move it to the systems in their data centers as needed. Now, on the client side, one of the notions that is sort of central to JavaFX is two things: number one is depending on the device you're on, the browser makes a lousy interface. Number two is a browser is typically a hostile environment. So and again, an example is I talked to a customer who builds a competing product in a space with Microsoft. And what happens is that when they're running in IE and they go ahead and type in a search term, they expect it to go to some service that exists or some data that exists on their site. Instead, it works them off some Microsoft-owned property instead. Again, why? Because the browser is the container. The browser can intercept and redirect as appropriate or how it determines or who its author thinks is the right thing to do for that search term. So you want to break out of the browser. So as a result of things like the latest version of the Java Runtime Environment which includes with it a new plug-in which allows for applets to be dragged outside of the browser such that [A] it provides a new deployment mechanism, but [B] it breaks you free of the browser environment. Number two is the fact that those applets can run JavaFX, since JavaFX is just layered, again, on top of Java, it can run on any device. We're moving forward on releasing JavaFX on top of MSA devices so that we can allow all those two billion handsets in the world to actually start using JavaFX and not require something which is inherently very expensive, especially in BRIC countries. And then you can take that application and move it between cell phone to TV to set top to desktop and get essentially the same experience as you move that device, across those different devices.

Hal:

Great. So if I'm a developer or I'm a system administrator – okay, I'll give you a two-part question -- what do I think about next? What's my starting point for thinking about how I'm going to incorporate the variety of technologies you just discussed?

Bob:

All right. So if I'm a developer, I want to start taking a look at FX. I want to start taking a look at what value we can provide, especially in the cases of providing rich media and content experiences, not only in the browser, but outside the browser. I want to take a look, especially if I'm a Java developer, I can start to recognize how FX as a language can leverage a lot of the power of the Java platform, but do it in such a fashion that allows me to build and deploy these experiences a lot quicker than I could by just programming directly to the Java language itself. I want to, as a system administrator, start taking a look at how I can start leveraging services. If there's ways that I can leverage virtualized environments in a more effective fashion and start thinking about how I might be using services in the cloud as opposed to – and sort of blend those with services that might exist in my data center, and this notion that I can actually move between those two depending on workload, depending on requirements, and depending on what my customers or my infrastructure required. And, again I want to take a look at how I can then bridge the gap and start using those services which exist not only in the data center, but in the cloud as well.

Hal:

Great. Well, Bob, as always, thanks for joining us on the show. I think a concise and really strong view of what's happening in cloud computing, adding some much needed grounding around a lot of things we see flying around, if you'll pardon the pun, in the trade press, and certainly an exciting time to be a developer right now. There's a new set of services, a new set of paradigms and hopefully a lot more capability underneath that to go get new code, new services out quickly.

Bob:

Thank you very much. It's a big area. It's very exciting. And obviously, you know, we're only spending a few minutes basically touching the edge of the cloud, if you will. And I'm sure that as things progress internally, we'll be seeing a lot more and a lot interesting details coming from the wide variety of folks within Sun.

Hal:

Great. Well, again, Bob, thanks for being on the show. You've been listening to another edition of Innovating@Sun and I'm your host Hal Stern.

Announcer You've been listening to Innovating@Sun. Join us next time for the latest in innovation from Sun Microsystems only on the Sun Microsystems Podcast Network.

[END OF AUDIO]