STINEMAN: Hello, everyone, and welcome to this IBM seminar, Building Flexibility into BPM Systems with WebSphere ILOG BRMS Decision Services.

My name is Brett Stineman, I'm the product marketing manager for the WebSphere ILOG Business Rules Management offering. And I'm joined today by Chris Berg, who is in the WebSphere ILOG Business Rules Management product management team. And, looking forward to giving you some very interesting information about what we're doing between the WebSphere ILOG BRMS products and the WebSphere Business Process Management product line.

Before we go into the session, I would like to provide one reminder. This session is being recorded. If for any reason you do not wish to be recorded, we ask that you log out of the session now. And with that, let's begin.

So to provide a quick agenda, I'm going to cover the first section which is really just focusing on the synergies between Business Rules Management and Business Process Management. At that point I'll hand it over to Chris, who will talk about the actual integration between our WebSphere ILOG JRules product and WebSphere Process Server. And then, he'll focus on some various use cases related to that integration.

And then we'll do a Q&A at the end, and you're welcome to ask your questions at any time during the session by using the chat button. And we'll also allow people to ask their questions on the phone.

So I believe most people who are listening today have some familiarity with Business Rules Management, but just in case you don't, let me take a minute to deliver the quick value proposition. So, basically what you can think of is there's decision logic that is scattered throughout organizations.

And it can be in many different places including directly embedded within application code, it can be within process models, it can be within various types of documentation, or even in the minds of people such as subject matter experts within an organization.

A Business Rules Management System allows you to centralize that decision logic into a repository, provide a set of tools that are specifically designed for various users who are involved in the Business Rules Management lifecycle. These can be developers, these can be business users and business experts, as well as IT operations folks, and having the tooling for each of those to be able to define, analyze, maintain and manage rule-based applications and the various business rules that are within those.

And then lastly, having an execution component that allows different systems to be able to access and utilize the decision logic that's stored in the repository. So on top of that, you want to be able to have some tooling within the actual, what we call the Rule Execution Server, to be able to manage the deployment, execution and monitoring of that decision logic that is being accessed by various business systems.

And the benefits you are getting from this, you can see in the upper lefthand corner. One is, you're eliminating those decision silos because you're moving that decision logic out of all those various places. You're making it accessible to both business and IT because you're using a language that both sides can understand and agree upon in terms of how it relates to various systems that it affects. And you can see an example of a business rule here in a very easy to understand and domain specific language.

And then lastly, you can implement very fine-grained and content specific decision logic, so you can make the business rules that are able to deal with, how do I want to deal with one specific customer versus the next customer, one specific transaction to the next, one process instance to the next as well as being able to enforce standardization if necessary. If I have some regulatory

requirement I can make sure that my systems are following those requirements.

So that's Business Rules Management in a nutshell. And a question that really comes quite frequently that I hear is around, so what's really the difference between process management and rules management? And I think both are really focused around helping to enable agility within the enterprise but each of them is going about it in a slightly different way.

So process management is really focusing on, how do I want the set of activities to occur that are part of what I do as a business? And this can be a combination of both activities that involve people as well as systems, helping to coordinate both of those as well as helping to coordinate different groups within an organization that need to work together as part of a process. So as you can see, it's really concerned around operational efficiency and coordination.

Rules management, on the other hand, is really focusing on, what do I need to do within specific points within a process? So it's focused on automating decisions at a fine-grained level, and it's also really looking at the operational intelligence of the organization. How do I make sure that I'm doing the right thing at different

points along the way?

Now, if I was to go kind of one step deeper in terms of Business Process Management and Business Rules Management we can see here some more specific definitions. So BPM, right, very focused on the flow of that process, making sure that the right set of activities happen in the right order. It has a lot of focus around making sure that people are involved at the right points and helping to coordinate those.

Looking across different organizational boundaries, departments, even internal and external boundaries. And it's really trying to give transparency to the process so that everybody can look at it and understand it and make sure that they're doing it in the right way. It's also looking at both very long-running processes as well as ones that may have some straight-through processing and occur quite quickly.

Now, Business Rules Management, on the other hand, is really looking at decisions from a data orientation standpoint and looking at specific points where a decision needs to be made and where you want to be able to automate those wherever possible.

And it can be used in a number of different ways, so it can

be used as part of a process but it can also be used as part of any business system that an organization has. And whereas BPM is really focused on process-oriented transparency, Business Rules Management is focused on decision-oriented transparency. So it's helping to give that visibility of those decisions that are driving the critical business applications within an organization.

And in many cases it's helping to enable straight-through processing, because wherever possible you want to be as efficient as you can, and the Business Rules Management System can look at specific points and determine whether a scenario can be a pricing decision, eligibility decision, et cetera, where you want to be able to go ahead and move to the end point of that process and conclude a transaction or provide the customers with what they're requesting.

So one of the things that I mentioned in the previous slide was around promoting reuse using Business Rules Management. So I think this is quite important in the fact that you can take that decision logic that is being managed within the Business Rules Management System, you can deploy it out as what we call a decision service that can then be utilized by any process within an organization, within any other system.

It can be a legacy system, so we have ways of making those

decision services made as COBOL code, for instance, that you could actually embed directly within a legacy COBOL application. But basically, any system that would need to use that type of decision logic can access it from the Business Rules Management System.

And the benefit you can see here is one, you can help streamline processes because you don't have to embed all that individual decision logic within the process model, but you can also facilitate change because you can change one place and you can make it available across all the various systems that need it.

And you can also maximize automation, because what you're doing here now is being able to define that decision logic in such a way that is much more difficult in kind of the traditional ad hoc approach to embedding it in all the various places.

And then lastly, you're improving governance because now you have a repository, different groups can go in there, they can see what that decision logic is, they can determine if it's correct and they can also monitor where it's being used in the production environment. And we have tools that allow you to go back and look at that and determine what particular rule was executed for a particular transaction, for instance.

So looking at a specific customer we have, in Europe, there is an insurance company in the property and casualty insurance business. And they are an excellent example of using these two technologies together. So they looked at their claims process, and what they wanted to be able to do was basically unify how they're managing their claims process across the Nordic region in the various countries that they work within in the Nordic region.

So they began with a BPM-related effort and specifically using WebSphere Process Server, and they saw some good benefits from doing that, but they realized that there were a lot of decisions that were being embedded within that process and they wanted to be able to externalize those, manage those as corporate assets, basically.

And because of that, they realized they needed a Business Rules Management System and they chose WebSphere ILOG JRules to do that. And then they basically brought WebSphere Process Server and WebSphere ILOG JRules together as part of this overall claim system.

And the benefits they saw from doing this from their first project, which was in the single country in Finland, was decreasing the cost of their claims process, so they're able to decrease their costs there, as well as being able

to turn around claims much more quickly and thereby increase customer satisfaction.

And then thirdly, because they were able to automate more decisions than they had been able to previously, they were able to create substantial savings in terms of the amount of work that employees were doing and able to make sure that employees were involved in those specific situations where there's real value add in having a subject matter expert involved in a claim adjudication process.

So in terms of the use of Business Rules Management you can see here a number of different ways that they were using that around validation at the beginning of the submission process, doing automation of decisions around liability and compensation as well as payment calculations, and then determining very early on whether a claim could go through a straight-through processing path and go straight to accounts payable for the payout.

From kind of a higher level, some of the other benefits they were getting from using a Business Rules Management System are being able to manage and automate that decision logic even when it varied from country to country. So, you want to be able to have a kind of standardization where possible but there are going to be differences because each country has its own specific set of rules and regulations

and laws that they need to follow and they need to be able to ensure that they're meeting those country-specific regulatory issues and using the Business Rules Management System to ensure that.

And then lastly, they wouldn't be able to accelerate the implementation of changes that they needed when they wanted to change their payout policies or they wanted to be able to react quickly to a new law that occurred in a specific country, for instance. They were able, using the Business Rules Management System, to do that much more quickly than they had before.

So now that I've given you the quick overview around this particular session, I'm going to turn it over to Chris Berg, and he's going to take you into some specifics around Business Rules Management, specifically the WebSphere ILOG BRMS offering and the WebSphere Business Process Management offering and how those integrate together. So, Chris, you should have control now, and I'll let you take it from here.

BERG: Thanks, Brett. Hello, everyone. As Brett mentioned, my name is Chris Berg, and I'm a product manager for ILOG JRules. And what we're going to talk about today is essentially, or in the second half of the presentation, we're going to look at some of the details of our integration.

And to highlight this, I'm going to focus on the areas of our synergy. And one of the best ways to look at the synergy is by reviewing the lifecycle because it's here that we see a lot of things happening. So Brett started the discussion around the difference in orientation, but another dimension or another perspective is to look at it just in terms of lifecycle. And we'll be uncovering this or diving into different dimensions in the next couple of slides.

But initially, the lifecycle really helps to define what the synergy is, because it's in the lifecycle that we discover where the contracts are between what's occurring in a process and what's occurring in a decision. It also introduces what the different roles are.

What are people doing, how are they working together to create their projects and then deploy them into an operational environment? And it also lends some information or it gives us some visibility into frequency: how quickly are things changing? So let's take a look at these in order.

On the BPM side, typically you have an integration developer who is using WebSphere Integration Developer or WID. And they are focused on creating their process model.

And those typically change about once or twice a year, sometimes more frequent. It does vary by client.

However, when you compare that to the lifecycle of a decision, they tend to occur more frequently. And whereas the role on the BPM side was an integration developer, as the role over in the Business Rules Management System is more likely to be either a business analyst or a developer that's working closely with the business such that they understand the very details of the data itself.

So if you recall the difference in orientations that Brett was talking about, one being process focus, the other being data focus, it's possible to create a process model but without knowing all of the data points that are part of the contract, but it's not possible to create that decision unless you really know what's going on in the data itself and understand the business.

So when you're doing your projects together, you've got different roles that you're engaging. And as Brett already described, what you're doing is you're using these two technologies to extend the reach of your overall solution. So you're reaching out, you're going after efficiencies from a process perspective, but then that automation is not only getting gains in terms of operational efficiencies...

But you're also expending the tool set out to participants in a project that might not otherwise be included or perhaps included only through a requirements or an interview process. So we're bringing more roles together into an operational environment.

Here's another view of a basic integration use case. And typically what happens when you are using the products together, there is a data model. The data model might begin in WID or it might preexist either project. In a lot of cases, there's already an application in case and there's a model. There's data that's moving around, and there's an investment in that model.

So in that case, let's imagine that it preexists the projects, that model can be brought into a WID project and it can also be imported into Rule Studio. And while there's a shared model, projects can basically continue in parallel and the process can be created within WID.

And in the case where the model is imported into Rule Studio, an architect or developer would work with a business analyst, they go would go through the process of making that model more understandable to a project. We call that the process of verbalization. So they're taking things that might not be immediately recognizable to the business and they're calling it something that the business

understands, the customer order.

And once they do that, once the project has been verbalized, it's brought into Rule Team Server, and from there the business analyst would typically continue to author rules. And even before they're done, they can deploy a partially completed decision through Rule Execution Server.

And at that point, while the signature of the decision has been created inside of the project, that is suddenly available at the time they publish the RES, that signature is available as a Web service and that can be imported into the WID project.

And then over in Step 4 here, the project can, you know, if they wanted to start before they had a decision available, they could create a stub for that decision, or they could immediate...if they knew a decision is available quickly they can grab the WSDL from the Web service, get that into the project and literally at that point both projects can be working in parallel and they can start iterating and maturing what they're doing in their respective projects.

Now, what happens when you bring a decision into WID? This is just a simple diagram to show that when you bring a decision into WID and you tend to deploy the WPS, what

you're really doing is you're creating a service component and it can do or have access to many of the same features inside a WPS that you would expect a business process to have or human tasks or business state machines. So we're really coming in at an SCA level inside of the environment.

And we do this in two ways. In the first case, we can integrate via Web service. So as I mentioned before you can pull a WSDL either from Rule Execution Server and/or an administrator can also make that available through UDDI. And that WSDL can then be used to create the decision in WID.

And it's basically acting as a service wrapper, the Web service that we provide acts as a service wrapper around Rule Execution Server itself. We support a lot of different ways to integrate. But this one in particular is what I would call WID centric, and it's going to give the basic functionality of the decision into the process model. And by doing that, of course, it's creating a service component.

And another way to this integration is basically a no code integration. So it's using native features inside of WID. Again, it's grabbing the WSDL. This is just one or two steps. In this case the two steps I'm showing here are pulling from it the file system. And then once you import

the WSDL it's going to create the interface for you.

And then it's literally a couple of quick clicks and you've created the implementation behind the service that you've just imported. You're going to choose the standard around SOAP. And once you click okay, you are off and running and you can immediately test the integration between WID and Rule Execution Server.

Now, a second way to do the integration is to use the decision service wizard. And this is currently available as support pack LA 71 from the IBM support site. And the URL for the support pack is on the final slide in this deck. And at the end of this session, we'll pause on that as well so that you can write it down or get a reference to it.

And when the support pack is installed, it provides a WID plug-in. And what this does is it provides a richer integration with Rule Execution Server, and rather than pulling the signature of the decision from WSDL, it's going to pull it from a portable representation of a decision.

So we have an archive. It's a jar file. And there's going to be a series of screens that one goes through. And very similar to the process for importing a WSDL, one can import a decision but with different features.

So, for example, rather than going through a Web service interface, it provides more direct access to the RES execution API. And it also provides out of the box integration with Common Event Infrastructure or CEI so that you can start using more features inside of WPS when you're using the products together.

And just to give you an example of the three screens that one answers when integrating a decision, in step one you're basically looking for an archive. An archive may contain multiple decisions -- in this case, I'm choosing data validation.

In step two, you pointed to your model, in this case the shared XSD that can either be part of the existing project in WID or you can pull it from another source. And then you answer some questions that help create the implementation behind the decision.

And at the last moment, in step three at the part where it says, implementation type, this is where we have some choices, so we can choose a POJO invocation, which is a direct invocation of the RES as a JCA. We can also do a direct invocation of the...I believe, excuse me one moment here. I can also do a direct call to a local EJB. And then I can also do a call to a remote EJB.

So there's a lot of different, there's at least three different session types here, and you basically choose them based on the way that you are deploying Rule Execution Server. And in a few slides down, I will go through some of the decision making process that one would go through when you're thinking about your implementation.

I won't linger on this slide. But what I'm showing here is an example of integration with the CEI. So this is a browser view of the common base event browser. And at the top of this are traces from executing decisions from within WPS.

And so what is happening is Rule Execution Server, every time a decision is made when a choice to include a trace is clicked, the decision trace is passed into WPS and can be tracked here. And as you can see, it includes things like the number of rules that were fired and so on and so forth, the version of the decision that was called.

And at times, this can be quite useful because at the time that you integrate a decision, whether it's through WSDL or through the decision service wizard you have a choice, and they're related again to lifecycle. So if you want a process instance to always have the same decision for that version of the process model, you can lock it down and say,

I only want this version of the decision, which then would allow one to create versions of decisions that are not ready for a specific version of the process.

Maybe you're working on a new process and it doesn't have all the historical clients on it and but you want to roll it out slowly, then you can create a new process instance bound to a specific version of a decision. In other cases, you may not care. You may always want the latest decision for all your process instances. In that case, we support that as well whether you're using the decision service wizard or WSDL.

And finally, if you come back to your history and you're interested in a specific process instance and you're wondering what version of a decision was used, you can look that up here through the trace information.

And finally, let's go into some of the use cases and practices. So earlier on I gave a very basic use case. And we're going to get into some more involved use cases here in a moment. But first let's look at choosing between Web services and the decision service wizard. So when I use the term HTDS here what I'm really referring to is the wrapper around Rule Execution Server.

So there are some differences here. So in the case of HTDS

or Web services, you need to have an XSD based model, and it turns out that the decision service wizard also supports that. But if you don't have a model as an XSD, the dec service wizard also supports a POJO model; and of course, HTDS supports SOAP and the decision service wizard is going to support the rule session APIs. And in that case, you look down here at the EJB support that's included.

And then also the common event infrastructure is bound to the service components that are generated by the wizard. But in both cases, there are ways to deploy this in a clustered environment, and there are some options for high availability.

Now, the decision service wizard has some deployment options here. And I mentioned before that there's basically three different session providers that can be picked. And I wanted to include just a quick reference here for when you would choose one over the other. They're all providing dynamic updates of a decision via RES, but some are supporting clustering and others are not.

What's nice about the EJB support is although the decisions themselves might not actually be transactional, they may participate in a transaction. So if there's complex calculations required to determine whether or not a calculation is valid, you can do that. And of course,

there's only one option for remote RES with remote EJB.

In other cases, many times folks are asking the question, well, when do I use a decision that's in JRules and when do I use a decision using rule logic artifacts inside of WID? And I typically address this issue again through the lifecycle. So if it's clear that the lifecycle of a decision is going to be independent of the process lifecycle and/or has different stakeholders, then you're likely going to want JRules to manage that.

Or in the case of, if you have multiple BPM implementations, that's another great use case for using JRules as a way to standardize your decisions. However, if the decisions are much more utilitarian or they are clearly bound to the process lifecycle itself -- meaning they're typically modified by the folks that are using WID, in integration developer, for example -- then you should use those rules.

And once you get into perhaps 50, 100 rules, then that's when you're going to want extra tooling around what you're doing. In many cases as you scale up the number of rules, that's also another way of saying that that decision is becoming more important to the business and it may start incurring some of the pressure points around having a different lifecycle, having a different way to manage

governance, different stakeholders, so on and so forth. And then of course, if you know that your decision is going to be effective across different platforms, then that's another great case for the use of JRules.

And, as I promised here's another view of the lifecycle. And again, this is just slightly more detailed than the basic use case I presented earlier. But what I'm trying to do here is just talk a little bit more concretely about where things are occurring between the two products. So I'm going to have a slightly greater focus here on the Rule Studio side, but essentially what I'm trying to do is show where things occur.

And so, we have in the first step a shared model, that model is being import in step two to Rule Studio. And while it's in Rule Studio that is where the architect or developer is working with the business analyst to develop the vocabulary.

And then after they've developed that vocabulary together, then that gets published in the Rule Team Server. And while it's in Rule Team Server, that is where iterations occur with the rules typically, that's where the governance is managed for the rule lifecycle. And while that is typically iterative, at any point it can be deployed to RES and be represented inside of a process model in step five.

So that's just a slightly more detailed view of that.

Now, there are more advanced ways to look at this as well. So if you know up front that what you want is more of a parallel way to do things, this kind of represents that. So there's a 2A and a 2B. In here we are recognizing that indeed it's likely that a project will want to work in parallel and the decision from JRules may or may not be available.

So in that case, if Business Modeler is involved, a service might be represented as just an activity with no implementation. Or if JRules is not yet installed or available, there might be some other kind of Java based implementation that's defined inside of the process with a given contract.

And then at the point in which the decision is available from RES, that implementation is removed and replaced by the decision service from RES. And what I call this, is kind of service abstraction, but it's a way to kind of virtualize the way that you're integrating the two projects. And it's giving you some agility in your project because you can say, well, I don't necessarily have to have the exact decision up front, I can work my projects in parallel and integrate them when it makes sense.

And I put this slide here for reference only. And basically giving some background about our support model for the support pack. In the case of JRules 7.0.2, you'll notice that it is showing Eclipse sharing. And what we mean by this is essentially that Rule Studio with the release of 7.0.2 can be used within the same instance of WID.

So you can easily change perspective in the same tool. You can move from creating process models to editing rules if you're a developer, for example. It's a great enabler for someone that's using that tool as their primary authoring environment.

And here's the bottom line: better patterns of reuse, so we're gaining agility simply by having a kind of a reference to these decisions used by multiple clients. Reduced time to market: so in the case of the customer in Finland, they're able to handle the pressure points that they have in their business across international boundaries, across many different pressure points that they have in their business.

They're getting better alignment because now they've extended their tooling to all these different roles. They're getting direct input from the business. And it's perhaps richer than what they were getting through a

traditional requirements process. And they're gaining, and I like the word "scaling," their ability to collaborate around requirements, because they not only can express them in something that's meaningful, but they can express them in something that's operational that they can use directly in a system and make it executable.

And they're also getting more consistent use of language so that the investments are made by the business to take those shared models and make them understandable to the business, that's going to help them scale and get better collaboration around their decisions to get buy-in, to get transparency.

And they're also going to go after a larger set of complex problems. So in the case of greater...if they're going after greater efficiency in their business processes, now they're suddenly going to be looking at ways to leverage SOA, ways to reuse that decision in many different parts of their architecture and their deployed application.

So it's really a multi-dimensional approach to the way that you would solve a business problem when you're using these two products together. And Brett, I'll open it up to you, if you have any final comments of your own.

STINEMAN: I guess I would just urge people to take a look

at some of these reference links that we've provided. There's some great information here that you can use as well as the support pack that Chris talked about which actually allows WebSphere BPM customers to go ahead and start utilizing that integration with ILOG JRules.

We also have a trial version of JRules that you can access by going to the BRMS Web pages that we have on ibm.com. You can see those also on this page, and can start to work with the two products together. So I think that's something that a lot of people who are on this call will want to do.

I do have one thing I'd like to come back to, Chris, I think just as we're moving into the Q&A piece. One of the earlier slides that you had, you talked about this idea of, when do you use JRules versus when you might use the rules capability that's directly within WebSphere Process Server.

And I'd like to talk a little bit more with you about routing rules, because I think a lot of time people think of rules and processes as some sort of a routing decision: do I go Path A versus Path B? So can you maybe talk a little more about rules related to routing, and when you might want to choose to have those just directly within the BPM system versus BRMS?

BERG: Sure. Well, certainly the rules for routing are in WID and JRules doesn't provide that functionality. But another way to look at it is, when a rule is used for routing or a transition -- depending on the language you want to use -- the point is that is enforcement on the part of the process model to enforce the choice that's being made. However, JRules can still be used to set up the decision point prior to the transition.

So for example, in the case where there are multiple BPM implementations in place and only a single technology is desired to make the decision, that can still be done and then the different BPM platforms can then enforce in their own unique way those transitions or that routing within the system.

So you can imagine then that prior to each transition in the process model that you're going to see a JRules decision that's setting up for the transition. So that's one example.

And another way to view it is there's, a heavier weight decision on the part of JRules, and there might be then two or three different transitions that would occur in the process model that are feeding off of that data that's been brought into the context of the process.

STINEMAN: I think that's an important point because you may come to a point in a process where the number of different paths that that particular instance might need to go down is more than just A or B. And in that particular situation, this is where having a Business Rules Management System can be quite valuable.

We actually have some customers that think of their overall processes as a number of subprocesses that may need to be kind of connected together using the rule management system, because of the complexity that may occur at these various transition points that Chris is talking about.

Another thing I'd like to ask you about is around traceability of decision changes when you have processes that have inflight instances, and how do we deal with that particular situation.

BERG: Well, there's a couple of different ways to deal with traceability. So, certainly when JRules has used standalone we have the decision warehouse where we can monitor and persist the traces that are occurring at execution time. But when we use the product with WPS, and what we have the option of doing is passing those traces, as I said, into WPS and then it takes responsibility for managing and persisting that information.

And there's lots of different things that can be done with it, but one is to just simply add an extra layer of monitoring around the decisions that are taking place, such that if one ever had to go back and review a complete flow of a historical process instance, then they would have enough of an artifact to do that.

So it's really here a choice of, how are we fitting well with kind of that infrastructure and the kinds of traceability requirements that are occurring with the process. And I'm not doing justice to WPS on this point, but there are lots of ways to make this rich, much richer beyond basic CEI integration.

STINEMAN: Okay. There was a question that came in about looking at the different ways that you can implement this integration and performance metrics, so basically kind what are the performance implications for using the direct call versus the Web services based integration?

BERG: Well, some of this is based.... You know, performance issues are largely based on the deployment models that the clients will have. So the clients that I've spoken with that are using the support pack, they favor the decision service wizard for reasons of performance because when they look at what they're gaining through the Rule Execution Server API, they're giving a lot

more choice, lower-level protocols to connect with the service. And I'm comfortable with that.

In the case where they already have a predominant use of Web services, then that fits as well, and they're perhaps less concerned about managing the protocol of SOAP. So it's a mix. It's largely determined by what choices the client has already made.

STINEMAN: Okay. A couple other questions came in, I think you actually already answered as part of the presentation. I think what I would like to do at this point is, if people want to ask some questions directly on the line, if you press star 6 that will unmute your line, and why don't we open it up to a couple of questions.

QUESTION: Hi, this is Ellen, and I'm just wondering, is there a full process that can be set up to manage or govern the changes to rules?

BERG: Yes, that's an interesting question. And there is a sample in JRules that kind of points to this, on how to do this. So essentially there is a way through Rule Team Server customizations where you can take the interfaces that are provided and hook them up with the WPS, for example.

So you would have to hook them up. Basically it's a manual approach to hooking them up, but it is possible and the sample does show the integration points that you would use to do this.

So right now inside of Rule Team Server, the governance is enforced around the state of the rule set. So there are rules that are defined, they have a certain amount of privilege. And as the decision goes through different states of rule that one person has may be able to deploy but another one may not. But where you want to have a kind of governance process, then you can create a process and basically enrich what's already there through a process.

STINEMAN: Yes, and that sample that Chris is talking about is part of the JRules 7 release. So, in case you're looking for it, that's within JRules 7 that that exists. I think related to what you're talking about in Rule Team Server so obviously we can enable...there's different people who have different roles and responsibilities and capabilities in terms of what they can do within Rule Team Server.

Also, the status of the rule itself can be used as part of the governance, so the ability to have different status of where that rule is and whether it's new or defined or deployable, et cetera, or deployed, and those can also be

customized.

QUESTION: So if I needed to have two or three business people approve whatever business rule before it was allowed to even go into like a Q/A process.

STINEMAN: So you could use that as part of Rule Team Server and enable different users to be able to set the status of each rule. And so a certain person may only be able to say that that rule is deployable or can go to QA, for instance, as part of the check.

QUESTION: That would be dependent on...or, that would have to be set for each business rule, then?

STINEMAN: That would, yes. And I think if you have more questions on that, we may need to take that offline and get back to you on that.

BERG: Yes, just a quick follow-up. It could occur at different levels, so it could be at a rule level, and/or there are ways to manage this at a decision level as well. So there's lots of choices. So let's move on to the next question.

QUESTION: Hi, could you just discuss a little bit more about how the BPM calls the BRMS, and then how the decision

in the workflow is utilizing the result of the call to the rules engine?

BERG: Sure. At a high level what's happening is every decision that you create in JRules has a data contract with it, so we call that, that's the data model. So a decision might require customer object, it might require an order. It might be a case, it might be a policy, an insurance policy -- it's whatever your business requires. So that becomes part of the signature. Now, it's likely that your business process also needs the same object.

So at the time that you integrate a decision with your process model, what you're doing is you're establishing what data is going to be part of that interface inside your process model. So there is interfaces inside the process model, there's interfaces with those external services that you're going to bring in to the process. And again, they're all sharing the same or similar data contracts.

So in the case of JRules, you're going to send it in an instance of customer data, and it's going to return either the same model with some additional information that says, customer is approved for a loan, or customer belongs to a campaign for X. And then that information can then be used inside the process.

So up to the point in the process where the decision is called, it's going to pass information to JRules, you're going to get some different data back. And then that data is now available in the context of the process model. So in the case of WID, it's available in SDO, and I can create more logic around that data inside of the process. And if needed, I could send it to yet another system to participate in some kind of transaction.

So what you're really doing is you're taking that business logic, putting it in a place where it can be managed by the business. And then you're taking those decisions -- the results of those decisions -- and you're using them in your process. So that's kind of the short answer.

STINEMAN: I was going to say one other thing following up on that. So, sometimes the rule engine will receive all the data it needs from that invocation from the BPM system. There may be other cases as part of the rule set that it's processing through that it may need some additional information.

It may need to make another call to pull information from some other data source in order to complete that rule set, and then pass back the output parameters that are going to go back into the BPM system into that particular process

model and process instance.

BERG: So you can think of a decision as kind of a function: you're going to pass data in and you're going to get data back out. In some cases the data is overwritten with new data; or, in other cases, there may be a completely different response from the [decision]. And that's up to those that...it's up to the requirements of the business to decide what that signature is going to be.

STINEMAN: I think we have time for one more question, and then if there's other ones that have come in, we can take those offline and respond back.

QUESTION: Will this presentation be made available online after the call?

STINEMAN: Yes, absolutely. So as with all of our sessions, we will provide an e-mail with a link to the recorded presentation and to the actual slide deck that people can look at.

QUESTION: Okay. Thank you.

QUESTION: Is there any case study that's related to the healthcare insurance industry?

STINEMAN: There are actually some of those. I think we

have some customers in the healthcare industry. We have actually several of the Blue Cross/Blue Shield affiliates, use the JRules products. We have some other customers using our BRMS, our .NET product in the healthcare insurance industry.

I think we have to respond back separately just because we'll have to provide you some links. So I think you sent in that question via chat, so we'll just respond back to some pointers with specific examples of customers in the healthcare industry.

QUESTION: That would be very helpful.

STINEMAN: Okay. Well, I think at this point we're at the top of the hour. I'd like to thank everyone for joining us today. And again, we will send out a follow-up with recording of this session. So, thanks again.

[END OF SEGMENT]