**First Topic:** When to use to a rules engine vs. machine learning—Capital One's experience (AMER)

To know when to use a rules engine over machine learning and vice versa we first need to know what business logic, calculations and implementing business logic. Quoted from the speaker “Business logic is business decisions, things that govern a business process”( Bonham, “Rules Engine”). They are a set of business rules that change according to what the market needs at the moment. Think of business logic as a dynamic if-then statement. The speaker then says “calculations are used by business logic and they tend to stay the same” such as calculating probability, net income or assets. These are usually calculated the same. Business logic uses calculations to output information according to the business rules. Both business logic and calculations are brought together in a workflow, an example of a workflow from the speaker is an orchestration. There are different ways to implement business logic, application code, decision table, rules engine, etc. Application code is familiar to developers and easy to reproduce although one downside is that it is inefficient from a business standpoint. In the ever-changing market different rules would be added some might be changed and some will be removed. It is inefficient because the developer that implemented it would have to keep going back to change the code in anything happens to the business logic. In a decision table it is easy to add a new rule into a table without having to change surrounding code. Decision tables have better flexibility than application coding. The speaker says that “a rules engine removes some of the complexity in a decision table” generally speaking, of the three stated, the most efficient for implementing business logic would be a rules engine. Although it ultimately depends on use case, if you need to calculate grades, you do not need an entire rules engine to do that and a simple application code would be the best to use. Moving on from the rules section we will now go to machine learning.

Artificial intelligence (AI) is set of systems able to perform tasks that normally require humans and machine learning is a subcategory of AI that provides the ability to automatically train the system such as a product recommendation system, spam filtering, adaptive pricing systems, etc. Basically, rules are usually human defined rules that are set in an rules engine and information is fed into the engine which then outputs according to the rules defined. A machine learning system also uses machine defined rules. Rules and machine learning can go hand in hand in different ways. Based on information provided we can use a machine learning system to output into rules, we can use rules to output into a machine learning system, we can also use rules to output into a machine learning system which then outputs into another machine learning system. In conclusion, going back to the first question when to use Rules vs. ML? Use rules when the exact logic is known, and it does not need to derive the logic from the data. Use machine learning when the exact logic is not known but some inputs may be known, and it has lots of data that an algorithm can use to define logic. A human always creates rules, unless it is used in the method mentioned earlier, and machine learning software generates the code using algorithms is learned from training to create a machine learning system.

**Second Topic:** Managing Red Hat Enterprise Linux 8 for SQL Server database administrators (AMER)

SQL Server is available to deploy on Linux operating systems, bare metal servers, virtual machines and containers on OpenShift platform. Complete compatibility across all platforms as stated by the speaker “a backup of SQL Server taken from a Windows operating system which is then restored onto SQL Server running as a container in OpenShift platform and it will continue to function just fine” (Shah, “SQL Server”). Installation is native based on the OS, package installation and update/upgrade using yum for RHEL. SQL Server service runs natively using systemd. If you are familiar with SQL Server command-line tools, there is a Linux native version of these tools available to download. Why would you want to put SQL Server on Linux OS such as RHEL? Another speakers says “when you choose to run SQL Server on RHEL, you’re choosing the world’s leading commercial database on the world’s leading commercial Linux platform with the ability to run it on a leading cloud platform”(Imershein, “SQL Server”). RHEL Ansible Automation combined with RHEL system roles provides a simple yet effective way to automate several key tasks of your application, the solution is in human readable format and being agentless it is more efficient and secure. Included with the RHEL subscription is the new and improved Red Hat Insights. The speaker Don says “RHI strengths SQL Server and RHEL by offering several SQL Server specific recommendations to proactively detect issues and prevent system risk before they occur”(Pito, “SQL Server”). Simply put RHI is about keeping RHEL systems operating as smoothly as possible.

**Third Topic:** A practical example of how change data capture can transform a business (AMER)

“Events are driven by the characteristics that are required by modern applications or information systems”(Broudoux, “Change Data”). The four most important capabilities right now are agility, isolation & availability, real time and multi-clouds. With agility they are looking for loose coupling between applications to allow for continuous upgrades. With isolation & availability resilient deployments with isolation to enable high availability is required. With real time, users should expect a real time experience in modern apps. Finally, with multi-cloud deployments it requires a robust communication layer for synchronizations. All of these capabilities make your application more robust and adaptable for changes. To change data capture with debezium they first deploy a Kafka broker and configure a debezium connector to listen to our existing database. The debezium connector listens to the binary logs of the database and produces 3 events corresponding to the 3 tables in the database. The events produced by debezium are basically a json message with payload snapshots before and after the change occurs. To filter out specific data and get information from users they decided to use Quarkus, a super-fast framework, they believe that they will need it to process millions and billions of events. The next step is using the Red Hat integration platform as a service called Fuse Online. This platform leverages the syndesis upstream project and it allows the design and deployment of integration services with no code into a cloud-based integrated environment. For the integration service to start interactions with customer through telegram they used the new service called Camel K. With Camel K you just have to develop your service into a single file, and you use the Camel CLI to push it to your Kubernetes cluster. On the server side the Camel K operator will take care of building your component as a container and deploy it. In conclusion, stream processing offers a complementary development model to correlate events. Camel can then be used as a swiss-army knife to bring the various connectors.

**Fourth Topic:** Why pay Red Hat for open source-making the case for paying for free software (AMER)

How does an enterprise obtain and manage software today? They either build it themselves, buy it, download it from the community, or purchase a subscription for enterprise software. Some of the advantages and disadvantages of building it themselves are that is has the most flexibility and has a low initial out-of-pocket cost. The cons are the productivity will be limited by headcount, resources are not dedicated to the business, and they hold the entire responsibility of maintenance, security, and upgrades. For buying the software outright the pros are that you can keep your developers focused on the business, the application is maintained, and security is outsourced. Cons are that the innovation happens at your suppliers’ pace and is dictated by wallet share, it is limited flexibility and transparency of functionality, and the threat of vendor lock-in. For downloading & deploying straight from the community the pros are the fast innovation, low acquisition cost, and the possibility for low maintenance cost. The cons are that the software is sometimes a bleeding edge innovation, no hardware, software or security certifications, the responsibility falls on you for maintenance and security, the unpredictable level of community support that moves at the pace of the community, and the supply chain may not be secure. For purchasing a subscription for enterprise open source software, the pros are as follows, collaborative innovation and problem resolution, leverage the speed of community innovation with tested certified and hardened code from a secure supply chain, ongoing support the security and maintenance is outsourced, and the transparency and influence over roadmaps. The author of this presentation has a hard time laying out the cons because this is the best option to choose.

The speaker states that “a Red Hat subscription is all about access and advocacy” (Pacek, “Open Source”). The perks for access to a Red Hat subscription is that it is open source enterprise software, it supports emerging open source technologies, it has integrated monitoring, management and remediation, lastly it has a long-supported life cycle and great expertise. The advocacy on the visibility and influence you have over Red Hat product roadmaps, the customer needs in open source projects, and finally the advocacy over hardware, software, and cloud provider partnership.

In favor to the paid subscription some benefits of a Red Hat paid solution are as follows, the number of unplanned outages per year with a benefit of 30% over unpaid, MTTR in hours has a benefit of 43% over unpaid, hours of lost productivity per user per year has a benefit of 63% over unpaid, and the value of lost productivity per organization per year with full time equivalents has a benefit of 63% over unpaid. The 3-year cost of operations per organization is $20.24M for unpaid solutions and $13.09M for paid solutions.

Citations:

Bonham, Andrew, and Edoardo Vacchi. “When to Use to a Rules Engine vs. Machine Learning—Capital One's Experience (AMER).” *Red Hat*, 29 Apr. 2020, onlinexperiences.com/scripts/Server.nxp.

Imershein, Louis, Don Pito, and Tejas Shah. “Managing Red Hat Enterprise Linux 8 for SQL Server Database Administrators (AMER).” *Red Hat*, 29 Apr. 2020, onlinexperiences.com/scripts/Server.nxp.

Broudoux, Laurent. “A Practical Example on How Change Data Capture Can Transform a Business (AMER).” *Red Hat*, 29 Apr. 2020, onlinexperiences.com/scripts/Server.nxp.

Pacek, Dan. “Why Pay Red Hat for Open Source-Making the Case for Paying for Free Software (AMER).” *Red Hat*, 29 Apr. 2020, onlinexperiences.com/scripts/Server.nxp.