CENTERVIEW™ ARCHITECTURE

WHITE PAPER

Overview

There are several questions that any architect or technology advisor asks about a new system during the evaluation process:

- How will it fit into our organization and technology stack?
- How will it fit with our security model?
- What other systems will it be inter-operating with?
- What are the touch points to other systems?
- What kind of resources will it require?
- How extensible is it?
- Is there an architecture diagram?

This paper is meant as a quick, architectural overview of Corda CenterView™ and will try to answer all of these questions and other common questions. This paper is intended for a technical evaluator or system architect and therefore, assumes a fairly technical background. However, it does not delve into the fine details of implementation. If more details are needed in specific areas, please contact Corda directly.

Corda CenterView Components

Corda CenterView is made up of several applications:

- CenterView Builder is a desktop application written in Java, which is used to design and create the dashboard applications. Because it is a Java® application, it can run on Windows®, Mac OS/X®, Linux® and other operating systems for which Java is available.
- CenterView Server is a server application written in Java which runs in a Java web application server. This is the main CenterView application which serves up the dashboards as browser based applications.
- CenterView Administrator is a server application written in Java, which also runs in a Java web application server. The Administrator is used to configure and manage one or more CenterView Servers. It is recommended that this application be installed on a secure server, different than the CenterView Server in production installations.
- Corda Mobile is a specialized mobile application written for the Apple iPhone®, which allows access to CenterView dashboards on the iPhone. Other mobile devices can access CenterView dashboards if they have web browser functionality.

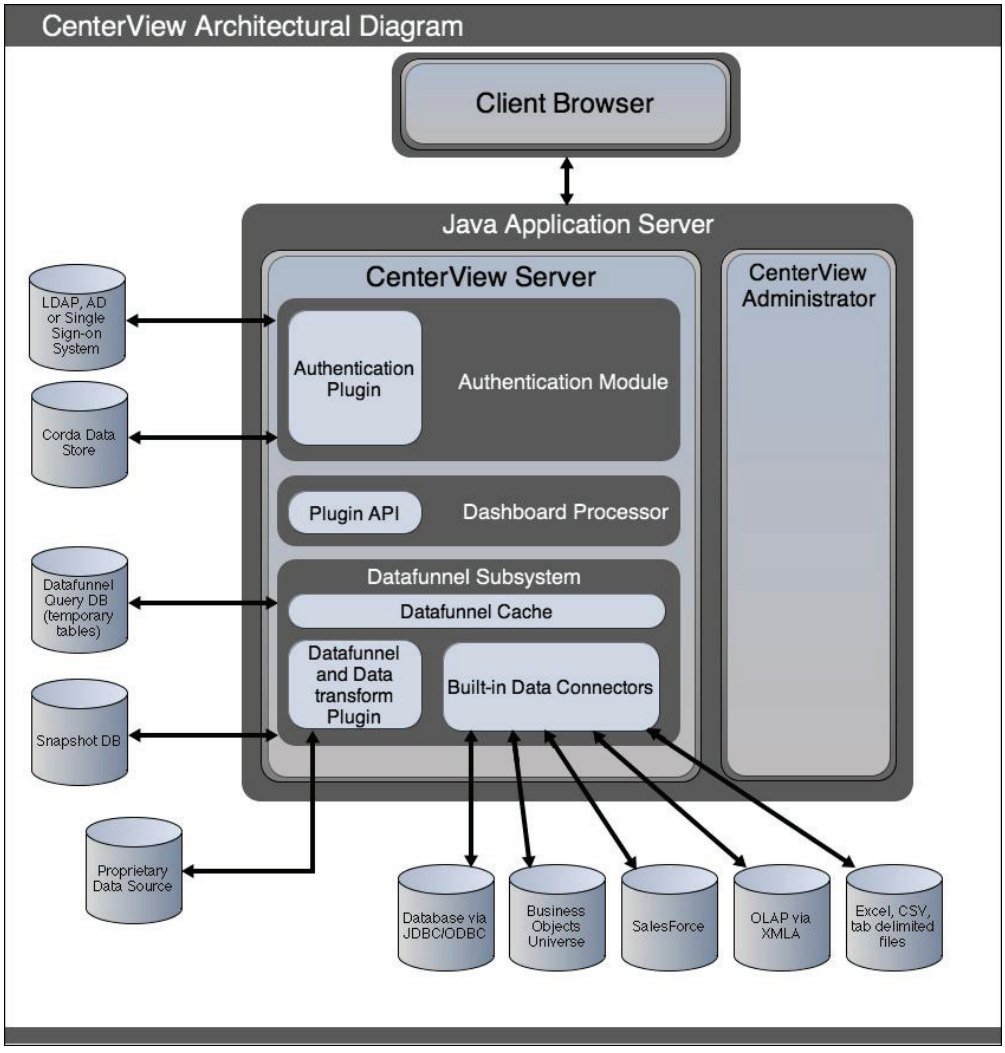
The majority of this paper will focus on CenterView Server and CenterView Administrator.

Integration

Both CenterView Server and CenterView Administrator are implemented as Java-based web applications (see "Architectural Diagram"). As such, they require a Java web application server environment to run. CenterView ships with Apache Tomcat™, which is an open source (reference implementation) Java web application server. CenterView can be deployed on the included Tomcat, but many implementors choose to install it using their existing Java web application server. CenterView can be installed in any of the major Java-based systems such as Oracle WebLogic®, IBM Websphere®, Oracle Application Server®, JBoss® or Apache Tomcat.

Implementation in Windows-only shops that do not have a Java web application server environment may choose to run CenterView Server and CenterView Administrator in Apache Tomcat, and configure Microsoft IIS® to use the Apache Tomcat connector. This connector allows all traffic between the client browser and CenterView to pass through, and appear as though it is coming from IIS.

Any of these application server environments can be configured to use SSL as an added measure of security. This is especially important if the dashboard applications will be available outside the corporate firewall.



Supported Browsers

CenterView currently supports the following browsers:

- Microsoft Internet Explorer® version 6 and newer
- Mozilla Firefox® version 2 and newer
- Apple Safari® version 3 and newer
- Other browsers may also work correctly, but are not officially supported.
- Mobile devices can access CenterView mobile dashboards through their built-in browser. Mobile dashboards use a very limited set of HTML, CSS and no Javascript, so most of the built-in browsers work with this output.

Plugin Architecture

CenterView designers wanted to have an open architecture that could be extended as needed. Some of the specific areas that custom capabilities may be needed are authentication and authorization, loading data from proprietary sources, and the ability to create custom code to handle embedding of unique content such as Google Maps™.

CenterView is designed to support custom capabilities through plugin modules. A set of APIs and interfaces are provided to extend CenterView Server in specific areas. The custom code is written in Java (or a Java wrapper to native code).

Authentication and Security

A key part of any implementation is providing secure and restricted access to dashboard applications. While some applications may be available to everyone in a company, others may be restricted to certain groups based on their role.

CenterView can be configured to integrate into existing Active Directory®, LDAP or a custom authentication environment. Any of these systems can be configured to support SSO (Single-Sign-On) if the infrastructure is in place. Active Directory and LDAP authentication can usually be accomplished via configuration parameters only. Different authentication environments will require a custom authentication module, called an authentication plugin. An authentication plugin communicates with a database or internal system through an API and verifies authentication/access permission for the specific user, as well as the groups/roles the user belongs to and other key information for the user. The authentication plugin is written in Java. Often, SSO systems are quite unique to a company and vary from home-grown to customized commercial implementations. Integrating with these environments will also require that a custom authentication plugin be created. Several examples of authentication plugins are included with the product as a starting point for a custom solution. Corda Professional Services is also available to help with the creation of a custom authentication plugin.

CenterView Administrator is used to grant access to dashboard applications, pages within those dashboards and even individual KPIs (Key Performance Indicators) based on the groups/roles that a user belongs to.

For environments that do not have a directory service (or existing authentication solution) but still want to restrict access to dashboard applications, CenterView includes a built-in security system of users and groups that can be administered from the CenterView Administrator.

The information provided to CenterView in the authentication process may also be used when accessing data from databases and other datasources. This allows database access policies to be enforced for users of the dashboard applications when necessary.

More detail on security, including how CenterView guards against cross-site-scripting, SQL injection, variable spoofing and untrusted access, may be obtained by reading the Corda CenterView Security whitepaper.

Data Sources

When CenterView Server loads data from an external source, it is represented internally in what is called a *datafunnel*. A datafunnel is a two-dimensional abstraction of the data, no matter what the source of the original data is. Once data is loaded into a datafunnel, it can be manipulated, transformed, filtered, reformatted and operated on in a variety of ways before it is consumed into a graphical representation (graph, map,gauge, etc.) or presented in another form, such as a table.

- CenterView Server can access data from a variety of sources. Database access is accomplished through Java JDBC drivers. In Windows environments, ODBC is also supported. Some of the popular databases that are supported include: Oracle®, DB2®, Microsoft SQLServer®, MySQL™, and PostgreSQL™ to name a few. But any SQL database for which there is a JDBC driver, should be accessible from CenterView via database-specific SQL queries. Accessing other SQL databases will require the library (.jar) files that contain the JDBC driver be placed in a location where CenterView Server can load them.

- CenterView Server can also access data in Microsoft Excel® files, CSV and tab delimited text files, and HTML tables from internal or external web pages.

- Data residing in Business Objects™ Universes can also be accessed directly by CenterView Server. Versions R2 and R3 are supported. Some configuration will be required to copy the appropriate Business Objects API libraries to a location where CenterView Server can access them to communicate with the Business Objects server. These libraries are specific to the version of Business Objects that is installed.

- CenterView Server includes a data connector for Salesforce.com®. The connection to SalesForce.com requires a username and password to gain access to the data. The user account can be one set up specifically for data access, or can be tied directly to the user viewing the dashboard application.

- OLAP cube data can also be accessed by CenterView Server via the XMLA interface. Cubes that support this protocol include Microsoft Analysis Services, SAP® Business Warehouse, Oracle Hyperion Essbase®, Pentaho™ Mondrian and others.

- CenterView Server also provides a plugin architecture to access data from sources that are not natively supported. This allows custom Java code to be written to access proprietary data systems. This same mechanism can be used to create compute-intensive data transforms that operate on data in an existing datafunnel. When data is accessed in CenterView Server, it is loaded into what is called a datafunnel. One or more of these datafunnels can then be further queried, including database joins, as a new data source. This allows queries to be performed across databases, and against Excel or CSV files. This is accomplished by turning an in-memory datafunnel into a temporary database table, and then the SQL queries are performed against one or more of these temporary tables. This process only happens when doing an SQL query against a datafunnel. Normal datafunnel transforms and manipulations are done completely in memory. This ability to create a temporary table from a datafunnel requires that a database server be configured in which these temporary tables can be created. CenterView Server ships with PostgreSQL which is a feature-rich open source database. However, the database used for these queries is fully configurable.

- The second unique data source is called the *snapshot* database. The tables in this database are created from datafunnels as well, similar to the datafunnel query tables. However, they are not temporary. They can be added to or replaced on a scheduled basis. This allows snapshots of points in time to be taken from data sets that are transactional, allowing trending information to be gathered over time. The default for this database is also PostgreSQL, but this also can be configured independently of the datafunnel query database.

Caching, Scheduled Queries, Scheduled Snapshots

Data loaded by CenterView Server is cached internally as a datafunnel in memory. This allows subsequent requests for the data to come from cache rather than requesting it again from the data source. This can provide a significant increase in performance when multiple users are accessing the same data. Caching from databases and other dynamic datasources is set to 15 minutes by default, but this default value can be changed. Also, each datafunnel can specify a specific timeout, which is either longer or shorter than the default. Data that is loaded from files such as Excel will time out when a different version of the file exists, or when the item in the cache is removed due to inactivity and memory constraints.

The amount of memory allocated to caching is configurable. Once the cache in memory is full, CenterView Server can optionally be configured to spool some cached items to disk in preference to being removed from the cache. Reloading cached items from disk is often much faster than going back to the original data source.

In environments where data sources are slow, or queries are complex, it may be beneficial to schedule datafunnel loads. This may be particularly helpful when data may update only daily or a couple of times during the day. A scheduled datafunnel load can take place during off-peak times and have a long timeout. As users access the dashboard throughout the day, the data accessed will have already been cached. This technique may also work when a larger set of data can be loaded, and data specific to a user can be sub-queried from the larger working set based on the user role (a data set with thousands of rows). This technique may not be appropriate if there are many users/roles and each user's/role's set of data is different, or the amount of data that would need to be loaded into the cache is very large (a data set with millions of rows).

Scheduled loads can be controlled based on time(s) of day, day(s) of week, day(s) of month, with granularity down to the second.

Datafunnel snapshots can also be scheduled to happen at regular intervals. A key benefit here is the ability to take snapshots in time of transactional data metrics. Many systems can tell you the current state of the data, but cannot tell you the state of the data one day ago or one month ago. Snapshots allow you to create trending information on key metrics by saving off those key metrics in a scheduled datafunnel snapshot.

Alerts, Notifications and sending email

The CenterView Notification functionality sends email alerts and notifications to users using Sun's JavaMail API. CenterView Server does not include an SMTP server, therefore it requires access to an SMTP server to send the email messages. CenterView Server can communicate with SMTP servers that require authentication or secure connections (TLS/SSL). The SMTP connection port is also configurable. email addresses for CenterView users can be added manually or pulled automatically from a directory service, such as Active Directory or LDAP, or a custom database using the CenterView authentication framework.

Embedding CV content

Many customers create content in CenterView and then embed that content in other applications or containers. CenterView supports a variety of methods for accomplishing this.

- CenterView JSR-168 portlet
  Corda delivers a portlet that is JSR-168 compliant. When the Corda context is deployed within a compliant portal container, the CenterView portlet will be available to be embedded into a portal page.
- Support for Microsoft Sharepoint®
  Corda delivers a Sharepoint Web Part that can be easily installed into the Sharepoint application server. Once installed, the CenterView web part is available to be added to a Sharepoint page. The CenterView web part communicates to CenterView Server via http protocol.
- Other Applications
  CenterView content can be embedded in any web application using the Corda CenterView JavaScript embedder. The embedded content can be a specific KPI, a page with several KPIs on it, or an entire dashboard. This allows customers who have created their own internal portal, or custom application framework to embed CenterView content as a first-class citizen withother content and not link to a separate page to view dashboard content.

Linking to Other Applications from CenterView

A very unique capability of CenterView is the ability to drill down to other applications from CenterView content. Say, for example, you have a bar in a chart that represents current opportunities in a CRM applications such as Salesforce.com. You can click on that bar and get a list of the individual opportunities, and then click on one of those opportunities and go directly into that opportunity in Salesforce.com. The information that is needed to identify the selected item can be passed from CenterView to the external application. This capability is available for any browser-based application that can be passed parameters on the URL, or as POST variables.

Pulling Content from Other Applications into CenterView

One of CenterView's capabilities is the ability to embed content from other applications inside of CenterView dashboards. Typically this is done by using a URL object, or by creating an iframe in an HTML code block in a CenterView page. The source for this URL object or iframe is then specified as a URL to the external content/ application that is to be embedded on the dashboard page.

Resource Requirements, Scalability and Sizing

Corda is often asked, "how many simultaneous users can a single CenterView Server support?" There are so many variables that can affect the number of users and performance that giving a definitive answer is pretty tough. However, here is some information that may be helpful in planning.

If there is already a Java web application server environment in place, then it might be possible to install CenterView Server in that environment. Depending on user activity level and dashboard complexity, one processing core/100 users might be a reasonable estimate. A minimum of 512 MB of memory dedicated to CenterView Server in small installations, and 1 GB or more is preferred when there are several users. If the number of concurrent users is over 500, then certainly more memory may be beneficial.

If CenterView Server is the first deployment that will require a Java web application server, then a dedicated server would be a good choice. For a workgroup or small department deployment with minimal usage, a dual core server with 4 GB of memory will probably be sufficient. For a larger department, a quad core server with 4 GB of memory may be more appropriate.

Internal tests on a dual-processor quad-core 2.0 Ghz Xeon® servers with 4 GB of memory, 1 GB of which was dedicated to CenterView Server. In the test case, we had a separate dedicated database server for the data source, and the database queries were fairly simple so that the database performance had very little impact on the overall throughput. We simulated over 2000 concurrent users, which were fairly active, and still maintain an average page load time of less than 2 seconds. Dashboard activity consisted of a new dashboard page being requested, or a simulated drilldown into a KPI, or a popped up KPI about every five seconds. The dashboard used for this test is one of the example dashboards and can be found at http://centerview.corda.com/corda. Click on the link to *Retail Sales*. While this dashboard will be very different from any given customer dashboard, it allows you to determine how much more complex your pages/KPIs are.

Conclusion

This paper answers, at least at a general level, the key questions that are common from any technical evaluator or system architect that is considering how CenterView integrates into an existing environment.

Please contact Corda if there are specific areas where more detail is required or if there are questions in areas that were not covered in this paper.