ActiveNet Architecture

Change Logs

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Description | Author | Time |
| 0.0.1 | Created |  |  |
|  | Update Activenet Web Pools | Nan, Liu | 2015-12-07 |
|  | Update ACM Server | Nan, Liu | 2015-12-08 |
|  | Update ACM/Org/Activenetlog Database | Nan, Liu | 2015-12-09 |
|  | Update RServer & JReport | Nan, Liu | 2015-12-14 |
|  | Update Bulk Email Service  Update League Scheduling Service  Add Jasper Report Server  Remove Perfmonstats & Add AppDynamics | Nan, Liu | 2015-12-17 |
|  | Update Client Browser & EntryPoint | Nan, Liu | 2015-12-22 |
|  | Update Active Enterprise Integration | Nan, Liu | 2015-12-29 |
|  | Update EntryPoint; Sitecatalyst still using | Nan, Liu | 2016-01-11 |

# Overview

The following document shows the overall system architecture of the ActiveNet system.

* ActiveNet is installed in both the Las Vegas and Toronto data centers, for US and Canadian customers respectively. Some of the physical architecture is slightly different between the two data centers, but the logical architecture is the same, so the diagram below applies to both data centers.
* Some components are shown which are currently under development, and either not deployed in the data centers, or deployed only in a test  phase. These components and their dataflows are shown with dashed lines.

# System architecture



## Client tier

### 1. ActiveNet web application

**Description:**

The main ActiveNet application is entirely browser-based, using the following technologies:

* HTML/Javascript: Most of the UI is a traditional HTML interface with Javascript as needed.
* There are some newer pages which use Ajax / JQuery for a more responsive or dynamic interface.
* Flex: A few components of the UI are written in Flex, although these are being converted to JQuery.
* Java applet: ActiveNet uses Java applets for some of the POS and membership functionality, specifically to access locally attached devices.
* PDF: Reports

**Environment**:

* JRE 1.8 for Applet
* Browser for AUI & CUI
* Adobe Flash for Flex

**Relation**:

* http(s) ActiveNet Web Pools
* http(s) ACM Server
* http(s) Report Server

### 2. Locally attached devices

**Description:**

Various portions of the ActiveNet application have special support for locally attached devices:

* Membership card scanning (barcode or magstripe), via keyboard wedge.
* Magtek encrypted CC/Debit card scanner, USB attached
* Camera for capturing membership card images, via Twain driver and Java applet.
* Printing of reports and membership cards, via PDF output.
* Printing of receipts on POS, via PC ports and Java applet.

**Environment**:

* POS/Card Scanner/Printer/Camera

### 3. Unmanned entry point station

**Description:**

This application takes membership scans and opens entry points through a gatekicker. It is a locally installed Java application, which synchronizes with ActiveNet via web services, and has its own SQL database for operation when internet access is unavailable.

**Environment**:

* JDK/JRE 1.6(should upgrade to 1.8 from 16.0)
* SQL Database(Local SQL Server)

**Build/Deployment/Tools**:

* Eclipse/NetBean/IntelliJ IDEA
* Utilities\Ant\build.xml build-ivy.xml/ivy.xml
* .ini configuration manually
* standalone java -jar EntryPointUI.jar
* embed in browser EntryPointApplet(AUI)

**Relation**:

* http(s) ActiveNet Web Pools

### Other web service clients

A few of our clients have used ActiveNet's web services to build specialized UIs or applications, such as:

* Single sign-on
* Activity listing / search
* More elaborate portals

## Application tier

### Load balancer

The load balancer VIPs are not explicitly shown in the diagrams. However, every gray box containing servers represents a pool of servers fronted with an F5 VIP.

### 4. ACM servers

**Description:**

(Under development) ACM is being used as a platform for reworked consumer UI. This will change the dataflow in the following ways:

* Consumer pages will be redirected to the ACM servers.
* The ACM servers will have custom page types for the ActiveNet UI, which will obtain data via web service calls to the servlets.
* For pages for which a custom page type has not been written, ACM will proxy the page from the servlet, then apply its own styling (called the "HTML Consumer" page or "screen scraper").
* In addition to N ACM servers, there must be an ACM cache invalidation server.

**Environment**:

* IIS 6.0
* .Net Framework 4
* Window Server 2012

**Build/Deployment/Tools**:

* Virtual Studio 2012
* ACM\Deployer\Deployer.exe
* Utilities\ACMOperator
* Utilities\CUIDeployer
* Others?

**Technology:**

* ASP.NET
* C#.NET

**Relation**:

* http(s) Web Browser
* http(s) ActiveNet Web Pools

**Database**:

* ACM

### 5. ActiveNet web pools

**Description**:

Each ActiveNet organization client (org) has a dedicated SQL database in the data tier, and dedicated Java servlets in one of ActiveNet's web pools. Currently each web pool has two servers behind a load balancer, although more servers could be added to a pool if needed for a very high throughput registration mode. Each org assigned to a pool has a servlet installed on each server in the pool.

* Largest code base by far -- 80-90% of development has been in the servlet
* Servlet methods generate HTML & provide web service end-points
* Redundancy: Load balancing with sticky sessions: Most pools have two servers, although so far only the rush pool has been actually load balanced.
* Scaling: Orgs are assigned to pools; pool capacity is limited by memory, not by CPU.
* Most processing synchronous, but some is deferred to background threads

**Environment**:

* JDK/JRE 1.6(deprecated, upgrade to 1.8 from 16.0)
* ServletExec(deprecated, upgrade to jetty from 16.0)
* Windows Server 2012 (deprecated, upgrade to linux from 16.0)

**Build/Deployment/Tools**:

* Eclipse/NetBean/IntelliJ IDEA
* Utilities\Ant\build.xml
* Utilities\ActiveNetDeployment (vb)
* sdi.ini(automated in DE, deprecated)
* service.properties(future)

**Technology:**

* Java(Servlet, JDBC, JUnit, WSDL)
* JavaScript(JQuery)
* Html
* CSS
* Flex

**Relation**:

* tcp CacheControl(default 6000)
* tcp JReport(default 1397?)
* http(s) ACM Server
* http(s) EntryPoint
* http(s) RServer
* https AMS, OFS, AAS, Asset
* http(s) Web Browser
* https Proofpoint?
* http League Scheduling Server
* fs shared storage

**Database**:

* Org Databases for every organization (eg. kansascityymca)
* ActivenetSites for global data (eg. orgs, sites, systeminfo, etc..)
* ActivenetLog for application log

### 6. CacheControl

**Description**:

CacheControl is a simple Java application which implements cache coherence across the multiple servlets for a given site, so that when configuration data like an activity is edited on one servlet, all other servlets will have the same information.

* When one servlet updates a cached record in its memory and in the database, it sends a message to CacheControl containing a serialized Java object.
* This message is forwarded to all the other servlets which have registered themselves with CacheControl for that site; those servlets in turn update their cache.
* It is not currently load balanced, and would require some application work to make it load balance-able.

**Environment**:

* JDK/JRE 1.6(deprecated, upgrade to 1.8 from 16.0)
* Windows Server 2012 (deprecated, upgrade to linux from 16.0)

**Build/Deployment/Tools**:

* Eclipse/NetBean/IntelliJ IDEA
* Deploy Manually

**Technology:**

* Java(Socket)

**Relation**:

* tcp ActiveNet Web Pools

**Database**:

* No Database Access

### 7. Report servers

**Description:**

When a canned report is executed in ActiveNet, this is done by a report execution request to a separate load-balanced pool of report servers.

* Behind firewall
* Servlet presents report execution UI; sends report request; report server replies to servlet with HTML, PDF or CSV results.
* Stateless, no caching, so easily load balanced.
* .Net application built on Data Dynamic's Active Reports product.
* ~200 distinct reports.
* Lots of filtering and display options in servlet.

**Environment**:

* IIS 6.0
* .Net Framework 2
* Window Server 2012

**Build/Deployment/Tools**:

* Virtual Studio 2008
* Configuration Manually
* Others?

**Technology:**

* VB.NET

**Relation**:

* http(s) ActiveNet Web Pools

**Database**:

* Org Databases for every organization (eg. kansascityymca)
* ActivenetLog for application log

### 8. JReport servers(Ad Hoc Report)

**Description:**

When an ad hoc report is designed or executed in ActiveNet, this is done via a third party application, JReports.

* JReport application is a pure Java application
* Servlet report editing and viewing pages are JSPs copied from the JReport source.
* Servlet manages reports via RMI requests to the JReport application installed in a separate pool of servers.
* Development provides metadata ("business cubes"), as well as sample reports.

**Environment**:

* JDK/JRE 1.6(deprecated, upgrade to 1.8 from 16.0)
* Windows Server 2012

**Build/Deployment/Tools**:

* JReport\ANJReportDefinitionEditor
  + java -jar ANJReportDefinitionEditor.jar
  + run & edit reports & save
* Deploy Manually

**Technology:**

* Java(JReport Enterprise Server API)[<JReport Enterprise Server API & RMI API>](http://www.jinfonet.com/manualsrv/en/userguide/srvapi.htm)

**Relation**:

* tcp ActiveNet Web Pools
* http(s) admin console app
* fs Shared Storage for catalogs

**Database**:

* Org Databases for every organization (eg. kansascityymca)
* Jreport for Jsper Report

### 8.1. Jasper Report Server

**Description:**

Product JReport is certainly planning on EOL, because they wrote a ticket I just executed to turn off JReports for any org which never used it, but they will have to speak to EOL date.

Jasper Report will replace JReport.

**Environment**:

* JDK/JRE 1.8?
* Windows Server 2012? Linux?

**Build/Deployment/Tools**:

* Data Team handled

**Technology:**

* Java[<Jasper Report Server>](http://community.jaspersoft.com)

**Relation**:

* Data Team handled

**Database**:

* Data Team handled

### 9. Bulk email service

**Description:**

When an org user sends bulk email, it is handled by a separate bulk email service.

* Database queued -- servlet write job data to tables in org DB; bulk email service goes to all org databases and looks for jobs; updates tables with result information.
* One instance of the bulk email service must be executing per major version deployed; each instance only goes to databases of sites on that major version.
* Uses Proofpoint MTA

**Environment**:

* .Net Framework 2
* Window Server 2012

**Build/Deployment/Tools**:

* Virtual Studio 2008
* Configuration Manually (ActivenetEmailSender.ini)

**Technology:**

* VB.NET

**Relation**: None

**Database**:

* Org Databases for every organization (eg. kansascityymca)
  + BULKEMAILACKNOWLEDGEMENTS
  + BULKEMAILATTACHMENTS
  + BULKEMAILRECIPIENTS
  + BULKEMAILTASKS
* ActivenetSites for global data (eg. orgs, sites, systeminfo, etc..)
  + ORGS
  + SYSTEMINFO
* ActivenetLog for application log
  + APPLICATION\_LOG

### 10. League scheduling service

**Description:**

This is a separate web service, which provides generates a team schedule.

* It is stateless, and requires no database access, so is easily load-balanced.
* The servlet provides a large XML object which describes the teams, facilities (fields), timeslots and other scheduling rules; the service returns an XML object which describes the schedule.

**Environment**:

* .Net Framework 2
* Window Server 2012

**Build/Deployment/Tools**:

* Virtual Studio 2008
* Configuration Manually

**Technology:**

* VB.NET

**Relation**:

* http(s) ActiveNet Web Pools

### Deployment servers

There are three related applications installed on the deployment servers:

* The ActivenetSites portal is a browser-based application for internal Active staff, primarily support and development, allowing them to configure new orgs, view the system configuration, login to orgs, query data, and view performance history.
* The Deployment Manager is a desktop application used by IT to start and manage Deployment Engine jobs to create new orgs, or update existing orgs with new ActiveNet releases, as well as automating some other ActiveNet IT tasks.
* The Deployment Engine is a component which actually executes a deployment job, updating an org to a specific version.

Required connections:

* Net: Incoming Http for portal
* Net: ? for IIS admin automation
* DB: Org DB
* DB: ActivenetSites DB
* DB: ActivenetLog DB
* DB: Perfmonstats DB
* FS: All web servers & report servers for deployment

Deployment:

* No automation or documentation.

## Data tier

All ActiveNet databases are on SQL Server instances in the respective data centers. File storage is on shares on network attached storage.

### 11. ACM database

**Description**:

Although all of the data displayed through the ACM-based consumer UI comes from the ActiveNet org databases, ACM does require a separate configuration database.

**Environment**:

* SQL Server 2012

**Build/Deployment/Tools**:

* Utilities\AnUpsizer (vb)(before 15.4[included])
* Kitmanager(from 16.0)

### 12. Shared File storage

A number of shares are established on network attached storage, for the following purposes:

* Uploaded organization file-based data, including graphics, documents, and membership card images.
* JReports storage for catalogs, report templates and other data shared across JReport server instances.
* Database backups for download or copy-to-trainer.

### 13. Org databases

**Description**

Each ActiveNet org has a separate database, containing just its configuration data, customers, receipts, etc.

* These can be configured to be spread across multiple database servers, although currently, each data center has only one SQL Server.
* Servlet does make limited use of data in ActivenetSites database for common information, such as zipcode table, SMS carriers table, etc.

**Environment**:

* SQL Server 2012

**Build/Deployment/Tools**:

* Create new org manually
* Utilities\AnUpsizer (vb)(before 15.4[included])
* Kitmanager(from 16.0)

**Note:**

ActivenetStartup:

Here's the basic procedure for creating a new ActivenetStartup.bak,

based on consultings work in Burnaby36

1. Use the US portal to login to Burnaby36
2. Download a database backup
3. Restore the backup locally
4. Rename as ActivenetStartup, either during the backup, or afterwards
5. Upsize it with the current ActivenetERD.xml, with "Drop Obsolete Data" and "Add Row Version" options selected
6. configure the servlet IDE to run that database, and bring it up to the login page, then stop the servlet
7. Delete the ActivenetStartup.bak file
8. Load and execute the sanitize\_activenet\_startup.sql script
9. Update ActivenetStartup.zip

ActiveNetTrainer:

1. Restore current ActivenetTrainer.bak
2. Upsize it with the current ActivenetERD.xml, with "Drop Obsolete Data" and "Add Row Version" options selected
3. configure the servlet IDE to run that database, and bring it up to the login page, then stop the servlet
4. Backup to ActivenetTrainer.bak
5. Update ActivenetTrainer.zip

### 14. ActivenetLog database

**Description**

Key ActiveNet components, including the servlets and report server, log their information to the application\_log table in a configured database.

* The application\_log table is currently in ActivenetSites, but the target model is for it to be in a separate database.

**Environment**:

* SQL Server 2012

**Build/Deployment/Tools**:

* Created manually
* Utilities\AnUpsizer (vb)(before 15.4[included])
* Kitmanager(from 16.0)

### 15. Jreport database

**Description**

Although the JReport ad hoc reporting system displays data from a specific ActiveNet org database, JReport also requires a separate configuration database.

* Each org's catalog and reports are registered in the database, although the actual catalog and report files are in the file system.
* No org data is stored here; reports connect to the org database to get their result sets.

**Environment**:

* SQL Server 2012

**Build/Deployment/Tools**:

* Created manually(include schema & initial data)

### 16. ActivenetSites database

**Description**

ActiveNet maintains overall configuration data for the data center, such as information on all the orgs and servers, in the ActivenetSites database in that data center.

**Environment**:

* SQL Server 2012

**Build/Deployment/Tools**:

* Created manually, use AnUpsizer to create schema
* Utilities\AnUpsizer (vb)(before 15.4[included])
* Kitmanager(from 16.0)

## Active enterprise integration

### 18. Enterprise services

The ActiveNet servlet uses the following Enterprise Services:

#### AMS

**Description:**

All credit card and ECP processing is done through the Active Merchant service. A diagnostic interface also uses AMS APIs to allow trouble-shooting of payment processing issues.

**Environment**:

* Client Side JDK/JRE 1.8

**Build/Deployment/Tools**:

* eclipse/netbeans/idea
* Configuration
  + Administration >> Financial Settings >> Configuration
  + Section : Merchant Account / Payment Processor Settings

**Technology:**

* Client Side JDK/JRE 1.8 CXF

#### Asset Service:

**Description:**

Any time certain products (activities, daycare programs and membership packages) are edited, the asset service is updated so those products will appear in A2 search.

**Environment**:

* Client Side JDK/JRE 1.8

**Build/Deployment/Tools**:

* Eclipse/Netbeans/Idea
* No Configuration : Schedule Task loaded from Org Databases

**Technology:**

* Client Side JDK/JRE 1.8 CXF

#### OFS:

**Description:**

At checkout time, a list of offers (magazines) is displayed based on data from the Offer Service; if an offer is accepted by the consumer, this information is sent back to OFS.

**Environment**:

* Client Side JDK/JRE 1.8

**Build/Deployment/Tools**:

* Eclipse/Netbeans/Idea
* No Configuration : Schedule Task related with Customer

**Technology:**

* Client Side JDK/JRE 1.8 CXF

#### AAS:

**Description:**

At checkout time, ActiveNet uses the Active Advantage Service (AAS) to determine whether a consumer is currently a member, to either discount fees or present Active Advantage sales information. If Active Advantage is purchased, this information is sent back to AAS.

**Description:**

At checkout time, a list of offers (magazines) is displayed based on data from the Offer Service; if an offer is accepted by the consumer, this information is sent back to OFS.

**Environment**:

* Client Side JDK/JRE 1.8

**Build/Deployment/Tools**:

* Eclipse/Netbeans/Idea
* No Configuration : Schedule Task related with Customer and loaded from Org Databases

**Technology:**

* Client Side JDK/JRE 1.8 CXF

### 19. Sitecatalyst

Each consumer web page will register data with Sitecatalyst, so Active can track the source of traffic to ActiveNet, and how many users stop at which point in the purchase process.

# Technologies used

The following table gives an overview of the various technologies used by ActiveNet for various purposes:

|  |  |  |
| --- | --- | --- |
| Technology | Vendor | Use in ActiveNet |
| SQL Server 2012 | Microsoft | All databases in the ActiveNet data centers, as well as in the Entry Point application |
| Java | Oracle/Sun | ActiveNet servlet (all of ActiveNet's business logic, as well as most of its UI) |
|  |  | Applets for a few ActiveNet UIs which require access to client resources, such as attached devices |
|  |  | Entry point application |
| .Net languages | Microsoft | ACM |
|  |  | CUI page types |
|  |  | Bulk email service |
|  |  | League scheduling service |
|  |  | Portal and deployment utilities |
| VB6 | Microsoft | DeploymentEngine, A few older utilities |
| HTML / Javascript / CSS | Open source | Most of ActiveNet UI |
| HTML5 / JQuery | Open source | Newer ActiveNet UIs which requirer a richer UI than static HTML can provide |
| Flex | Adobe | A few ActiveNet UIs which require a richer UI than was feasible with HTML when they were written. |
| JReport | JinfoNet | Licensed application for ad hoc reporting servers |
| ACM | Active | New consumer UI |