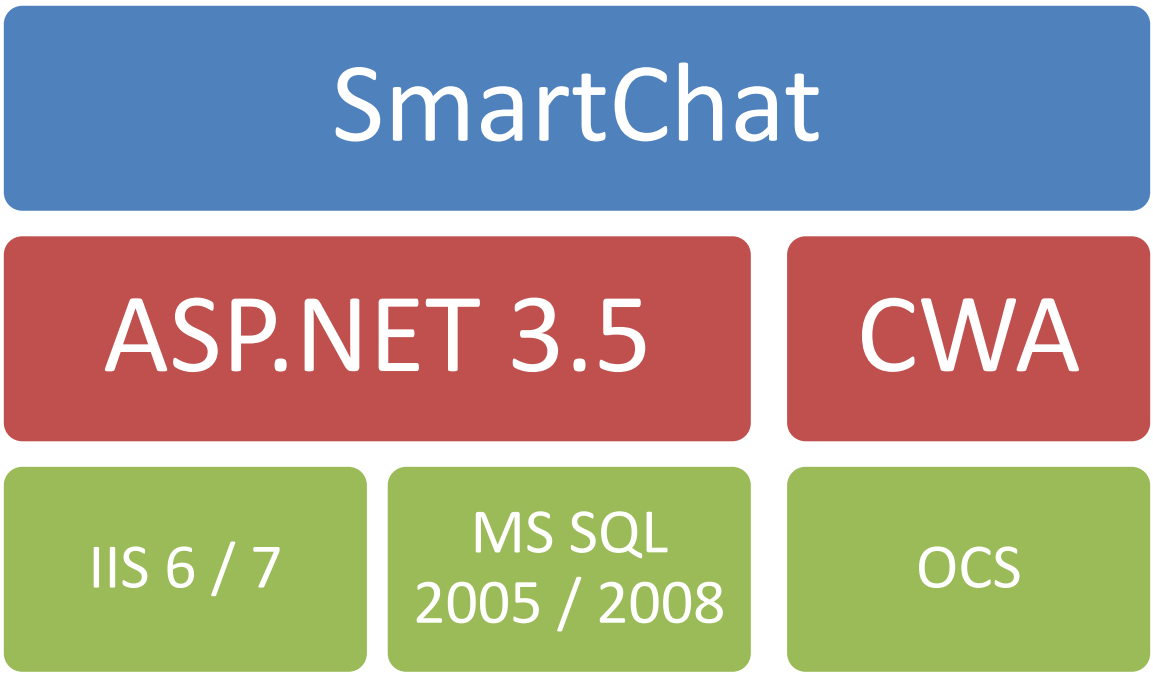
EC SmartChat : Deployment Whitepaper

Technology overview.

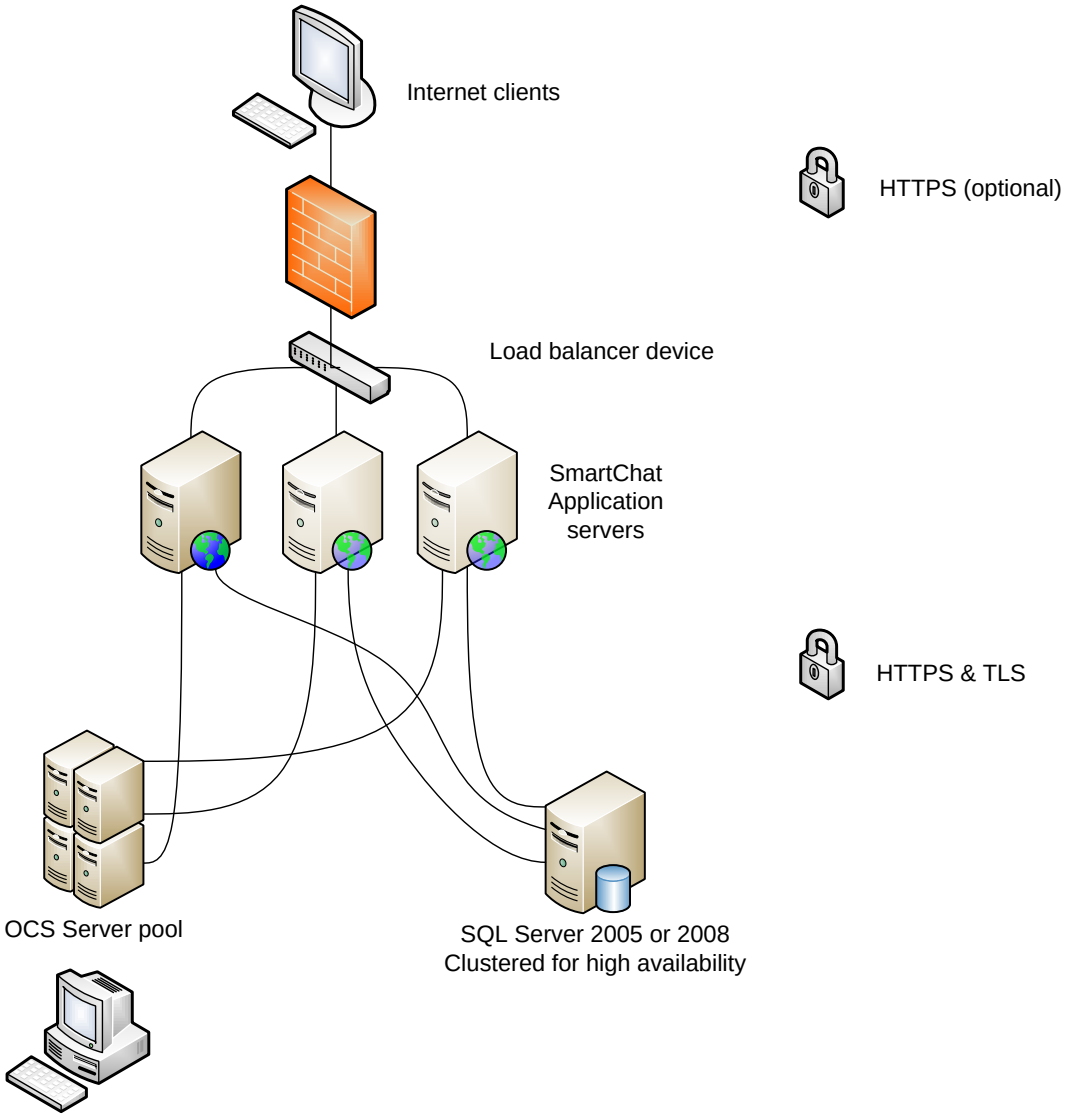
SmartChat is implemented as a stateless asp.net web application hosted on IIS 6 on windows 2003 or IIS 7 on windows 2008, with a MS SQL 2005 or 2008 server backend to preserve state.
The application can be scaled out at each layer of the application supporting millions of web clients.
The SmartChat application can be deployed either standalone on dedicated web application servers, alongside your existing web site or as an additional role on the OCS servers in the pool.
Global distribution is supported through the use of smart sub-domain redirection. The initial web request is made to the general address and subsequent requests are made to an alternative address.
State is preserved via the SQL server on the backend and can be replicated between sites either using SQL replication options or web services built into the application.



SmartChat technology stack

Scalable deployment options

The core SmartChat application is designed to be stateless which means it can scale out to any sized web farm to suit the load demand with basic load balancing equipment or software and no special configuration.
The SQL backend can be scaled using industry best practices for Microsoft SQL server for both performance and high availability.
The OCS infrastructure can also be scaled out as appropriate to support the additional anticipated load of this application.
For smaller installations, the application also supports co-installation either in your existing asp.net 2.0 (or greater) web application including MOSS or even in your OCS farm on any of the pool servers running IIS. The MS SQL database can be installed on either an existing SQL server or an instance of SQL express locally on the web server.



Typical deployment for medium sized farm supporting 18 million visitors.

Usage example: Chase.com

Daily traffic to Chase.com (compete.com, 9/3/2008): Unique Visitors : 18 Million
% Visitors to use contact center (messagecenter.chase.com) : 1% (alexa.com, 9/3/2008)

$$Users\ per\ minute = \frac{18\ million\ \times\ 1\%}{15\ \times\ 60} = 200$$

Assuming a 15 hr service window.
Assuming each visitor has an average 5 minute conversation, this is equivalent of an average of 1000 active concurrent conversations at any given time.
Our application generates a request every 3 seconds per active user = 300 requests / second.
NOTE : We are not affiliated with chase.com, we are simply using this publically available information as an example for scalability calculations.