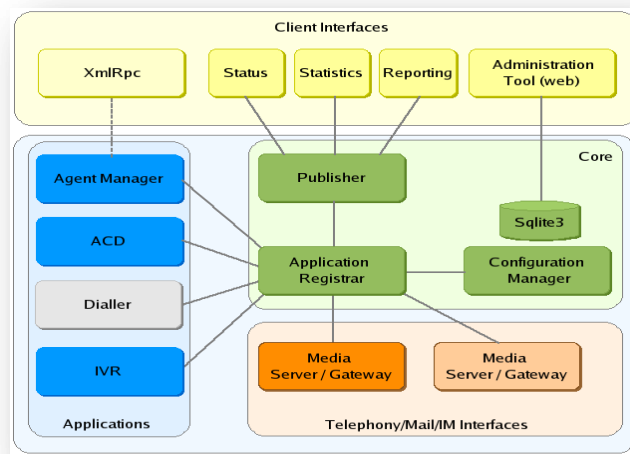


Application Server Architecture

ContactQ is an extensible protocol based Call Centre Application Server platform. The development of ContactQ has been driven by engineering standards including SIP, Voice XML (VXML), State Chart XML (SCXML), AJAX (Web 2.0) and XMLRPC. It has been designed to stand alone and for inter-working with VoIP Soft Switches. TDM hardware interfaces are supported through media gateways. The scalable architecture of the product allows it to be deployed both as a multi tenanted premised based solution and as a hosted solution. The product is available as a supported commercial product from Braxtel Inc, and an open source GPL project (www.contactq.org). Figure 1 below illustrates the system architecture.



Platform Features

- Deployed as CentOS appliance supported under commercial licensing terms
- Intelligent administration interface requires no Operating System knowledge
- Open Source core code base builds to GNU / Linux distributions (unsupported)
- SIP Gateway
- 6 ACD call routing modes
- Skills / priority based routing with 999 skill levels
- Cross platform desktop agent application
- API / SDK for user management and configuration
- IVR engine fully scriptable using SMXML
- Integrated IVR application graphical editor programming tool
- Historical reports with data archive management
- Real time reporting / dashboards
- Supports 100 Queues, 500 agents / supervisors
- Service gateway for HTTP requests enabling application integration
- Directory services integration and SNMP monitoring

* Whilst every effort is made to ensure the information given is accurate, all specifications are subject to change without notice.

ACD Routing Modes

Direct ACD routing modes are available in 6 variants for the ContactQ ACD and are configurable on a per queue basis. More advanced options are possible by routing calls through IVR applications.

Routing Mode	Description
First Free	Calls are routed to agents based on the order that they are assigned to a queue. The first agent assigned to a queue will get the first call and the second agent will be offered the second call, and so on. The first available agent will always receive a call unless unavailable, or if a call has been bounced where either of these conditions is met, the next available agent will take the call.
Circular	Calls are routed in a circular arrangement i.e. the next agent managed by the ACD.
Random	Routing to agents is performed randomly. Use cases for this routing algorithm might include training.
Calls Offered	Routing to agents is based on the number of calls they have been offered. The Agent who has been offered the fewest number of calls during this time will be offered the call first.
Longest Idle	The agent with the highest skill level for the queue that has been waiting in an idle state, for the longest period of time will be offered the call first.
Longest Between Call	The agent with the highest skill level for the queue that has not been offered a call for the longest period of time will be offered the call first. The main reason for this mode as an alternative to 'Longest Idle by Skill' is to stop agents abusing unavailable and non ACD activities as a way of resetting their idle time to go to the back of the queue

ACD Features

Feature	Description
Time of day Routing	Call routing can be adapted using scheduling, open, closed, holiday, temporary close combined with time and day call routing rules.
Routing directly via contact map to ACD or from IVR	Calls can be transferred intelligently from the ACD, either automatically or from the agent manager.
Multiple Routing Algorithms	6 Routing modes are supported (See ACD Routing Modes)

Braxtel™ ContactQ V1™ V1 Platform Overview Data Sheet

Multiple ACD Queues	Tested with 100 queues.
Agent skills	levels (9999 tiers) prioritises which agents receive calls in precedence to others
Automatic forced login	Agents can be logged in automatically from the ACD without the need for a management application or phone based login request.
Agent session management	Agent session management can be controlled through a phone, desktop contact application or API integration
Bouncing	Bouncing represents a trigger action when an agent does not answer an alerting call within a specified threshold. Media is presented to another agent when a media calls is bounced and a trigger action such as a change in queue presence state, or session termination (forced logout).
Call Disposition	Calls can be assigned disposition codes which have meaningful descriptions. Additional information can be associated with the call for the purpose of audit / reporting. Configurable disposition times ensure that agents are allocated sufficient time between calls to complete post call work.
Agent allocation to multiple queues	Agents can be allocated to multiple ACD queues with an associated skill level.
Call Announcement	An audible greeting is optionally played to an agent when they answer a call prior to connecting the speech path to the caller. This is an extremely useful option for agents who are logged into the ACD using handset management. Typically announcements would include the name of the ACD queue that they are receiving the call for. The option to play the greeting is configurable per agent.
Overflow between queues (3 modes)	<p>Timeout: This is the maximum amount of time that an unanswered call will stay in the queue. Once the timeout is reached then the call will be passed back to the dial plan for further processing by another application. The timeout can also be disabled so that the item is never overflowed.</p> <p>Call Volume: A maximum number of calls in the queue can be specified. Once the maximum is reached then one of the following rules are applied:</p> <p>Overflow New Calls. New calls will not be placed in the queue and will follow the overflow rules.</p> <p>Overflow the Oldest Calls. New calls will be put into the queue. The oldest call(s) will follow the overflow rules</p> <p>No Agents: This setting will determine if the queue overflows if there are no agents logged into the queue. Calls will be overflowed until at least one agent is logged into the queue.</p>
Call Logging	Blanket simple call recording to the file system on a queue wide basis.
CDR	Domain level records are written to a transaction safe cache for ACD calls events. The

events are written by the ContactQ Publisher application to the domain publisher SQLite Database.

IVR Editor and Features

The following features are specific to the ContactQ IVR application.

Feature	Description
Time of day Routing	Call routing can be adapted using scheduling, open, closed, holiday, temporary close combined with time and day call routing rules.
Multiple routing paths	Routing directly via contact map to IVR as well as from other IVR Applications or ACD Queues
SMXML Scripting	State Machine XML scripting interface integrated into application configuration. The scripting flow architecture is based on SCXML.
Media Control	Fully featured call control actions and parameters including but not exhaustively: Play, PlayRecord, Record, Conference, Application Transfer (ACD, IVR, External Address) and HTTP requests.
RTP Logging	Call logging can be set per call, with DTMF digit masking for security / compliance
Visual Editor	Integrated Visual Editor to Browser Application supporting Actions, Parameters, Responses
Application Templates	Application creation templates simplify the design of new IVR call flows. Defaults scripts include a simple auto attendant and agent routing by name or extension.
Outbound Email	Administrators and script writers can further enhance the IVR experience by creating and sending e-mail through the IVR. E-mail messages can accept attachments and example use case for this function could be voicemail to mail box.
HTTP Requests	Support for HTTP Requests enables script developer and administrators to make calls to HTTP applications and services, and utilise the response encoded as XML which can be parsed in SMXML and used in scripting for further actions such as playback, routing and email. A key benefit of this feature is that it allows IVR application developers to remotely extend the system's capabilities without adding expensive overhead to the platform's media processing capabilities. Examples include making remote service requests for secure and inexpensive database interactions, remote procedure calls such as XMLRPC, and interfaces into service gateways such as payment transaction processing and SMS messaging.
CDR	Domain level records are written to a transaction safe cache for IVR calls events. The events are written by the ContactQ Publisher application to the domain publisher SQLite Database.

API and Desktop Agent Features

Feature	Description
Agent Manager API	A low level ContactQ Protocol messaging interface is available for creating simple user level applications. This API is extended by an XMLRPC API.
Request/Response API	API requests are made over synchronous or asynchronous HTTP(s) channels using the request-response message exchange pattern.
Authenticated sessions	API sessions are authenticated in addition to user management for the platform. This ensures security and atomicity of client / server transactions and enforces permissible commercially licensed feature resource utilisation.
XMLRPC	The server level API is designed to be communicable using the XML-RPC protocol. XML-RPC allows API functions to be called from client applications over HTTP(s) with a discoverable registry option offering a self documenting interface into the services offered through the protocol called introspection. The XML-RPC server offers the ability to activate and manage its services to control endpoint location, security as well as the ability to use its HTTP capabilities to server files adjunct to the services offered by the API registry.
License based on concurrent users	The ContactQ API can be licensed for concurrent agents and authenticated API users.
Contact Application	The contact application provides a user experience for agents into ContactQ. It is designed to run consistently on a breadth of operating systems including GNU Linux, Mac OSX, and Microsoft Windows. The application is designed and built using the Adobe AIR / Flex runtime environment.

Historical Reporting Features

Feature	Description
Open Schema	A documented and published reporting schema enables adjunct reporting applications to be used for developing proprietary reports or intelligence driven views such as Key Performance Indicators (KPIs).

Braxtel™ContactQ V1™ V1 Platform Overview Data Sheet

CDR events	Domain level events are written to a transaction safe cache for ACD call and agent activity data records.
Reporting Application	An application interface reporting application is integrated into the user browser application.
Extensible report definitions	Reports are built from metadata definitions and report component associations configured within the reporting database enabling new reports to be defined without making application changes.
Archive Manager	Integrated Archive Management is available for reporting with a configuration Interface. This ensures that only the required level of reporting data is maintained in the current schema instance for performance and housekeeping purposes. Purged records can be archived into secondary reporting archive cache which can be restored and reported from on demand.
Export rendered reports to CSV	Once rendered, report data can be exported to CSV and downloaded locally from a web browser.
Report Templates	<p>The following report variants are configured in the reporting management database schema:</p> <ul style="list-style-type: none"> • Agent Detail, Parameterised on Agent and Date • ACD Call Detail, Parameterised on Queue and Date • IVR Call Detail, Parameterised on Application and Date • Inbound ACD Calls by Day, Parameterised on Queue, Date range or last week / month / year • Inbound IVR Calls by Day, Parameterised on Application, Date range, last week / month / year • Inbound Agent Calls by Day, Parameterised on Agent, Group, Date range, last week / month / year • Route Detail, Parameterised on Route and Date
User based assignment	Users can be assigned specific reports which can be run to utilise parameterisation constraints. This allows organisational partitioning of reporting data views including the ability for an agent user to be able to report on their own performance in relation to queue service levels.