VNGAGE INTEGRATION INTERFACE

Tracking and Monitoring

Technical documentation

1 Document history

Version	Date	Author	Note
0.5	2016-05-19	Bartul Bonacic	Created
0.6	2017-03-27	Fredrik Lunde	Case top level
1.0	2017-09-29	Stefan Möhl	First public version
1.1	2018-01-19	Stefan Möhl	Formalised message transcript Added visitor profile
1.2	2018-03-05	Bartul Bonacic	Added configuration.agent.collect command

2 Table of content

3 Introduction

3.1 Intention

This document aims at describing the integration interface provided by Vergic for associated partners and customers who wish to integrate their systems with VEP (Vergic Engage Platform). The intended audience is technical staff external to Vergic. It describes the system-level communication workflows and technical contracts.

3.2 Terms and Abbreviations

The primary users of the overall system functionality are the contact center agents, referred to as *agent* here. The other party to the agent is seeking contact and is referred to as *visitor* here. The dialogue of chat messages between agents and visitors is referred to as *conversation*. A competence group is a logical set of agents, usually sharing the same business competences. Such a competence group is referred to as *group*. A *queue* is an ordered list of visitors assigned to a group, often depending on the visitor's entry point. A CRM system is a Customer Relations Management system, keeping track of information known about a customer, including what interactions have occurred with a particular customer.

Abbreviation	Explanation
VEP	The Vergic Engage Platform
Partner System	The system connecting to VEP through this API
AMQP	Advanced Message Queuing Protocol is an open standard application layer protocol for message-oriented middleware
SSL/TLS	Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), both of which are frequently referred to as 'SSL', are cryptographic protocols that provide communications security over a computer network.
GUID	A globally unique identifier (GUID) is a unique reference number used as an identifier in computer software.
Agent	Contact center agents; the primary users of the overall system functionality
Visitor	The other party to the agent seeking contact from the web site
Group	A logical set of agents, usually sharing the same business competences

Queue	An ordered list of visitors assigned to one or more groups, often depending on the visitor's entry point or visitor attributes
Conversation	The dialogue of chat messages between agents and visitors
Case	An overall grouping of events, participants, time stamps and other meta data related to conversations

4 Business Use Cases

The purpose of this integration is to accomplish the business use cases described in this chapter.

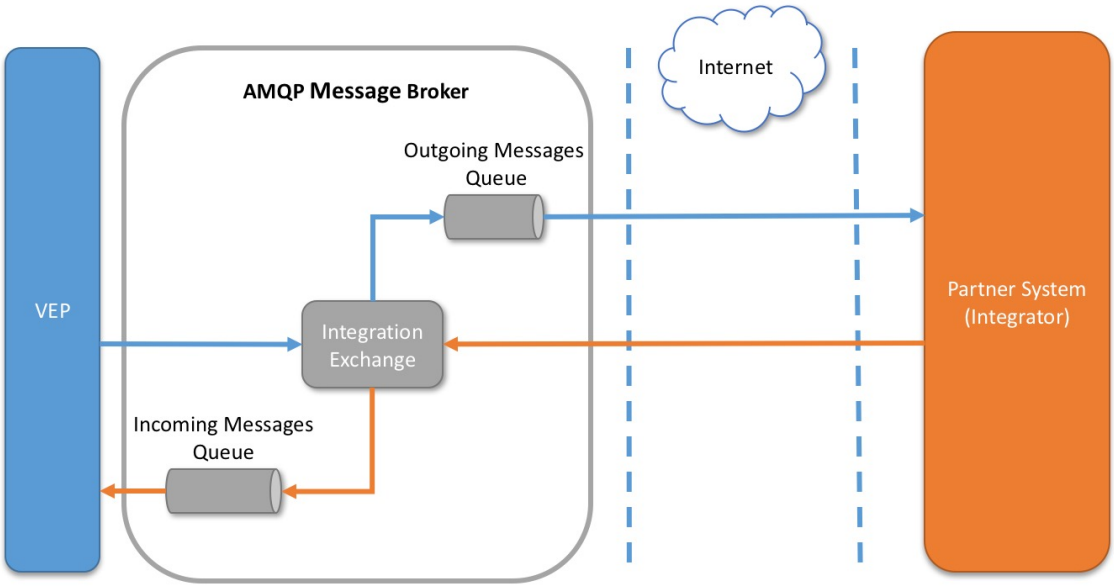
4.1 A CRM system needs to record case information

An external CRM system needs to receive a stream of records of cases in VEP, including meta data about the cases, and a transcript of the conversations.

4.2 Monitoring data about agents and queues

Statistics monitoring system needs real-time updates about the state and performance of agents and queues.

5 Technical Overview



The integration is accomplished via message passing between systems using a publish/subscribe system. Vergic provides a Message Broker to facilitate necessary communication between systems. This message broker communicates via the well-established AMQP protocol, providing a standard for transfer, acknowledgement, resilience and persistence of messages.

Events inside VEP that are of interest to the Partner System are published as messages and available for consumption by the Partner System in a dedicated queue marked as *Outgoing Messages Queue*, following the blue message flow in the diagram above.

Events inside the Partner System that are of interest to VEP are published as messages to the provided exchange (queue entry point) marked as *Integration Exchange*, subsequently available for consumption by VEP in the dedicated queue marked as *Incoming Messages Queue*, following the orange message flow in the diagram above.

6 Message Exchange Infrastructure

To realize AMQP broker functionality, Vergic currently uses a RabbitMQ server instance, accessible on a public endpoint available over SSL/TLS only.

The listener will register what events they will listen for and get an event stream that transmits the changed attributes of the underlying state.

The event stream may be deduplicated, and may be extended in the future, so the listener must be prepared for attributes not always appearing unless the event changed the attribute and also be prepared for extraneous attributes added as future expansion or for debugging purposes.

6.1 Publish/Subscribe

Messages sent over AMQP follow the publish/subscribe pattern. The sender of messages will publish a message on the message broker, however the sender does not need to know about the recipients (if any) of messages. The recipient of messages will subscribe on the broker to relevant topics from the publisher, whereupon events published on those topics will start to appear in a queue for consumption by the recipient.

6.2 Message Serialization

All messages are serialized into JSON formatted strings, more specifically - dates are serialized in ISO format and should always be in the UTC time zone; numbers as simple number values (decimal style); Booleans as ‘true’ and ‘false’ and all other simple types are created using standard JavaScript literals for the type. In order to successfully exchange messages, the Partner System must serialize messages in same fashion.

6.3 Event based

Messages are published to the AMQP broker as the specified events occur. The message will contain the attributes pertinent to the particular event, as specified in this document.

6.4 Message content deduplication

On some events, only part of the relevant state may have changed since the previous event of the same class. It is permitted, though not required that only the changed attributes are published. The first time an event type is published after subscription, there is no previously published data, so all attributes are guaranteed to be included.

6.5 Throttling

Some events may occur with a very high frequency, for example the queue state for a busy site may change many times per second when a large number of visitors enter and exit the queue, get picked up, conclude the conversation, and so on. For that reason, events are throttled to be produced no more often than at a configured rate. Intervening events are folded in to the events that are produced, and statistics are kept internally to ensure that every event is counted, even if the current state is only reported at a lower rate.

6.6 Extensibility

The messages and their parameter fields described below reflect the current version of the interface. Other messages may at any time be added to the interface, and other attributes may at any time be added to the messages, including temporary additions for debugging purposes. Because of this, it must be expected that messages or fields not mentioned here may appear and must be ignored.

6.7 Message Routing

All messages to VEP must be published to the AMQP exchange referred as *Integration Exchange*; the exact name will be provided by Vergic once the interface is configured. Each message must be tagged with the routing key of the data model name for the message sent. The Partner System is responsible for queue creation and queue binding.

6.8 Message Time-to-live

All messages have TTL of 5 minutes, which means that the message must be consumed by the Partner System in that timeframe or else it is lost. VEP will never retry message sending and will not keep track of non-routed messages.

6.9 Security

The Partner System must use specific authentication credentials and virtual host name to successfully connect to the message broker, all provided by Vergic once the interface is configured. Isolation of each integrator is accomplished by separating each on a dedicated virtual host. As the message broker is configured to use the SSL protocol exclusively when communicating over the Internet, the Partner System must use SSL as well.

7 Integration Model

7.1 Case

The case model has four main events: case.created, case.opened, case.closed and case.transferred. Case.created represents the start of a case, case.opened when an agent begins interacting with a case, case.closed when an agent has completed interaction with a case, and case.transferred when a case is transferred away from the current agent. When an agent closes a case, it may be marked as either resolved or unresolved. If resolved, it can be resolved as positive, negative or neutral.

If the full transcript of the conversation is required, a case.transcriptClosed event is available, occurring at the same time as case.closed.

7.1.1 case.created

When a case is created in VEP, a visitor has decided to interact with the VEP system. The visitor will have activated some opportunity on the web site, followed by the selection of an appropriate solution. One of the configured case types will also have been assigned to the case.

7.1.1.1 Routing key

When this event occurs, VEP will notify the Partner System by publishing following message with routing key – *case.created*

7.1.1.2 Message properties

Property name	Type	Description
eventTime	Date and time	The date and time when this event was created
caseId	GUID	A unique VEP identifier for the case
opportunityId	GUID	A unique VEP identifier for the opportunity
solutionId	GUID	A unique VEP identifier for the solution
caseTypeId	GUID	A unique VEP identifier for the case type

7.1.2 case.opened

Once an agent has been assigned to a case, the event case.opened is fired. The assigned agent will belong to a VEP group representing the business competences of the agent.

7.1.2.1 Routing key

When this event occurs, VEP will notify the Partner System by publishing following message with routing key – *case.opened*

7.1.2.2 Message properties

Property name	Type	Description
eventTime	Date and time	The date and time when this event was created
caseId	GUID	A unique VEP identifier for the case
agentId	GUID	A unique VEP identifier for the agent assigned to the case
groupId	GUID	A unique VEP identifier for the group

7.1.3 case.closed

When an agent is done with the case, the agent closes it. The agent may close the case as either resolved, in which case it will have an outcome of positive, negative or neutral, or close it as unresolved, in which case it will have an outcome of unresolved. A case closed as unresolved may be opened again by an agent at a future time.

7.1.3.1 Routing key

When this event occurs, VEP will notify the Partner System by publishing following message with routing key – *case.closed*

7.1.3.2 Message properties

Property name	Type	Description
eventTime	Date and time	The time when this event was created
caseId	GUID	A unique VEP identifier for the case
outcome	“positive”, ”negative”, ”neutral”, “unresolved”	One out of the set of four enumerated Strings, “positive”, “negative”, “neutral” or “unresolved”
closure	String	An free-text String describing the closure of the case

7.1.4 case.transferred.toGroup

A visitor may be transferred away from the current agent to a group for pick-up by another agent, for instance if the agent feels that another competence group might be more suitable to help the visitor, or if the agent's computer becomes unresponsive.

7.1.4.1 Routing key

When this event occurs, VEP will notify the Partner System by publishing following message with routing key – *case.transferred.toGroup*

7.1.4.2 Message properties

Property name	Type	Description
eventTime	Date and time	The time when this event was created
caseId	GUID	A unique VEP identifier for the case
previousAgentId	GUID	A unique VEP identifier for the agent previously assigned to the case
previousGroupId	GUID	A unique VEP identifier for the previous group
newGroupId	GUID	A unique VEP identifier for the new group

7.1.5 case.visitorProfile

When a chat is initiated, information about the visitor may be collected, e.g. through a web form. What attribute keys are in the visitorProfile depend on the configuration of the form.

7.1.5.1 Routing key

When this event occurs, VEP will notify the Partner System by publishing following message with routing key – *case.visitorProfile*

7.1.5.2 Message properties

Property name	Type	Description
eventTime	Date and time	The time when this event was created

caseId	GUID	A unique VEP identifier for the case
profile	JSON object	A JSON object containing a number of attributes and values

7.1.6 case.transcriptClosed

Once a case is closed, the transcript of the conversation becomes available.

7.1.6.1 Routing key

When this event occurs, VEP will notify the Partner System by publishing following message with routing key – *case.transcriptClosed*

7.1.6.2 Message properties

Property name	Type	Description
eventTime	Date and time	The time when this event was created
caseId	GUID	A unique VEP identifier for the case
transcript	JSON object	A JSON object containing a full transcript of the conversation that occurred while the case was open

7.1.6.2.1 Transcript

The transcript contains only the single property *messages*. Other properties may be added to the transcript property in the future.

7.1.6.2.2 Transcript properties

Property name	Type	Description
messages	JSON array	A JSON array containing all the individual messages of the conversation in the order they were received by the server

7.1.6.2.3 Messages array

Each element in the array is a message. Each message may have different properties depending on what type of message it is. All messages also have a number of properties in common. Please keep in mind that any property may be omitted if it contains no data, and new properties not mentioned here may be added at any time, sometimes temporarily for server debugging purposes.

7.1.6.2.4 Properties in common to all transcript messages

All messages in the *transcript.messages* array have some properties in common. For the sake of brevity, they are only listed here once, but the common properties are always included in every message of a transcript.

7.1.6.2.5 common properties

Property name	Type	Description
eventTime	Date and time	The time when this message was created
speaker	JSON object	The party that caused the message to be sent
recipients	JSON array	An array containing all recipients of the message

7.1.6.2.6 speaker and recipients properties of a transcript message

The speaker and recipient properties of the message in a transcript contain the GUID of the party as well as the external ID, if any, of the party. External IDs may be configured for an agent so that the agent may be matched in the partner system by the partner system's ID.

The speaker property is a JSON object with the following properties. The recipient property is a JSON array, where each of the elements in the array have the following properties. Each message may only have a single speaker but may have several recipients.

7.1.6.2.7 speaker and recipients properties

Property name	Type	Description
visitorId	GUID	The VEP identifier of this visitor
agentId	JSON object	An object containing the VEP identifier of the agent, as well as all external identifiers configured for the agent.
agentId.id	GUID	The VEP identifier of the agent
agentId.externalIds	JSON dict	A JSON dictionary containing a set of key-value pairs, where the key is the type of the external identifier, and its corresponding value. The keys and values depend on how the agent is configured.

7.1.6.2.8 Chat message

During a conversation, the agent and visitor may exchange text messages.

7.1.6.2.9 chat properties

Property name	Type	Description
messageType	"chat"	This is a message of type "chat"

message	String	The text in the message

7.1.6.2.10 Navigation message

During a conversation, the agent and visitor may co-browse. When someone moves to a new page during co-browsing, a message is sent with the URL that the speaker navigates to.

7.1.6.2.11 Navigation properties

Property name	Type	Description
messageType	"navigation"	This is a message of type "navigation"
url	String	The URL that the speaker navigated to

7.1.6.2.12 Note message

During a conversation, the agent may add notes to the case.

7.1.6.2.13 note properties

Property name	Type	Description
messageType	"note"	This is a message of type "note"
message	String	The text in the note

7.1.6.2.14 Link message

During a conversation, the agent may send pre-configured links to the visitor.

7.1.6.2.15 link properties

Property name	Type	Description
messageType	"link"	This is a message of type "link"
title	String	The title of the link
url	String	The URL of the link

7.1.6.2.16 Video message

During a conversation, the agent and visitor may initiate an audio/video conversation.

7.1.6.2.17 video properties

Property name	Type	Description
messageType	"video"	This is a message of type "video"
activity	"start"|"stop"	Either the string "start", when the video is initiated, or the string "stop", when the video is concluded.

7.1.6.2.18 Title message

During a conversation, the system may set a title to the dialogue.

7.1.6.2.19 title properties

Property name	Type	Description
messageType	"title"	This is a message of type "title"
message	String	The title of the conversation

7.2 Group

A group represents a set of business competences, and agents may be assigned to groups. VEP allows visitors to be placed in different queues; a queue is defined by having a corresponding group.

7.2.1 group.loadChanged

The event group.loadChanged occurs whenever agents or visitors change their status in a queue for a group. Each group will have a separate group.loadChanged message.

7.2.1.1 Routing key

When this event occurs, VEP will notify the Partner System by publishing following message with routing key – *group.loadChanged*

7.2.1.2 Message properties

Property name	Type	Description
eventTime	Date and time	The time when this event was created
groupId	GUID	A unique VEP identifier for the group
numberOfAvailableAgents	Integer	The number of agents currently available to receive a conversation
numberOfVisitorsInQueue	Integer	The number of visitors currently in queue for a conversation
numberOfVisitorsInConversation	Integer	The number of visitors currently in conversation with an agent

7.2.2 group.performanceTimers

group.performanceTimers is an event that occurs regularly, at the configured throttling interval, as long as there are visitors in queue or in conversation. The event allows monitoring performance-related timers of a group. Each group will have a separate group.performanceTimers message.

Though time intervals are given in milliseconds, this is for the sake of generality – it does not reflect the actual precision of timers. Since VEP is a web-based cloud solution, it is subject to network latency, browser performance, http overhead, personal computer and mobile performance at both the visitor and agent end, server performance during peak load and other factors, making the actual error on the order of a few seconds.

7.2.2.1 Routing key

When this event occurs, VEP will notify the Partner System by publishing following message with routing key – *group.performanceTimers*

7.2.2.2 Message properties

Property name	Type	Description
eventTime	Date and time	The time when this event was created
groupId	GUID	A unique VEP identifier for the group
averageWaitingTime	Integer	The average number of milliseconds that visitors have waited in queue during the last 15 minutes
longestWaitingTime	Integer	The longest time a visitor currently in queue has waited in milliseconds

7.3 Configuration

VEP can be configured to attach specific properties to different entities. For example, an agent ID in VEP may be associated with the name and email address of the agent configured in VEP, as well as the corresponding ID in an external system. The Partner System may subscribe to an event to be informed of changes to the configuration.

There may be several external systems mapped to a specific entity in VEP. These mappings will be reported as dictionaries of key-value pairs, where each key is the configured name for an external system (e.g. “Salesforce”), and the value is the corresponding ID for the entity in the external system.

Upon subscription, the entire set of configurations for the subscribed class will be published to allow an initial state, however, on subsequent updates to the configuration, it is possible that only changes are reported.

7.3.1 configuration.agent

Agents are the primary users in VEP, accepting conversations from visitors to the web site or other sources.

Upon subscription to this topic, all configured agents will be sent in separate messages. After that, updates to agent configuration will result in a message for the updated agent.

7.3.1.1 Routing key

When this event occurs, VEP will notify the Partner System by publishing following message with routing key – *configuration.agent*

7.3.1.2 Message properties

Property name	Type	Description
eventTime	Date and time	The time when this event was created
agentId	GUID	A unique VEP identifier for the agent
agentEmail	String	The configured email address for the agent
agentName	String	The configured name of the agent
agentDisplayName	String	The configured display name of the agent
agentAssignedGroups	GUID array	An array of GUIDs, one for each group the agent is assigned to.
agentExternalId	Key-Value dictionary	A table of key-value pairs containing { external system name: external system ID } for the configured external IDs of the agent

7.3.2 configuration.agent.collect

When Partner System requires the information of all available agents in VEP, it initiates command configuration.agent.collect toward this interface. After that happens VEP exposes requested data via previously described configuration.agent event, with an event per each agent.

7.3.2.1 Routing key

When this occurs, the Partner System will notify VEP by publishing the configuration.agent.collect message with the routing key – *configuration.agent.collect*

7.3.3 configuration.group

A group represents set of business competences that agents may have. VEP allows visitors to be placed in different queues; a queue is defined by having a corresponding group.

Upon subscription to this topic, all configured groups will be sent in separate messages. After that, updates to group configuration will result in a message for that group.

7.3.3.1 Routing key

When this event occurs, VEP will notify the Partner System by publishing following message with routing key – *configuration.group*

7.3.3.2 Message properties

Property name	Type	Description
eventTime	Date and time	The date and time when this event was created
groupId	GUID	A unique VEP identifier for the group
groupName	String	The name configured in VEP for the group
groupExternalId	Key-Value dictionary	A table of key-value pairs containing { external system name: external system ID } for the configured external IDs of the group

7.3.4 configuration.opportunity

An opportunity represents a measurement point in VEP, creating the option to take some action.

Upon subscription to this topic, all configured opportunities will be sent in separate messages. After that, updates to opportunity configuration will result in a message for that opportunity.

7.3.4.1 Routing key

When this event occurs, VEP will notify the Partner System by publishing following message with routing key – *configuration.opportunity*

7.3.4.2 Message properties

Property name	Type	Description
eventTime	Date and time	The date and time when this event was created
opportunityId	GUID	A unique VEP identifier for the opportunity
opportunityName	String	The name configured in VEP for the opportunity
opportunityExternalId	Key-Value dictionary	A table of key-value pairs containing { external system name: external system ID } for the configured external IDs of the opportunity

7.3.5 configuration.solution

A solution represents an action in VEP, to be taken upon some detected opportunity. Often, a solution will be to present a chat offer to the visitor.

Upon subscription to this topic, all configured solutions will be sent in separate messages. After that, updates to solution configuration will result in a message for that solution.

7.3.5.1 Routing key

When this event occurs, VEP will notify the Partner System by publishing following message with routing key – *configuration.solution*

7.3.5.2 Message properties

Property name	Type	Description
eventTime	Date and time	The date and time when this event was created
solutionId	GUID	A unique VEP identifier for the solution
solutionName	String	The name configured in VEP for the solution
solutionExternalId	Key-Value dictionary	A table of key-value pairs containing { external system name: external system ID } for the configured external IDs of the solution

7.3.6 configuration.caseType

Cases are grouped in case types, where the case type will be determined depending variuos criteria, for example the selected solution.

Upon subscription to this topic, all configured case types will be sent in separate messages. After that, updates to case type configuration will result in a message for that case type.

7.3.6.1 Routing key

When this event occurs, VEP will notify the Partner System by publishing following message with routing key – *configuration.caseType*

7.3.6.2 Message properties

Property name	Type	Description
eventTime	Date and time	The date and time when this event was created
caseTypeId	GUID	A unique VEP identifier for the case type
caseTypeName	String	The name configured in VEP for the caseType
caseTypeExternalId	Key-Value dictionary	A table of key-value pairs containing { external system name: external system ID } for the configured external IDs of the case type