Architecture of Conversational Platforms

Published 1 March 2017 - ID G00323532 - 30 min read

By Analysts Magnus Revang, Brian Manusama, Anthony Mullen

Supporting Key Initiative is Artificial Intelligence

Conversational interfaces will revolutionize how humans relate to machines, and EA and technology innovation leaders must understand this new paradigm to stay ahead. We look at chatbot frameworks and virtual assistants — VCAs, VPAs and VEAs — through the lens of a common conversational architecture.

More on This Topic

This is part of an in-depth collection of research. See the collection:

- Applying Artificial Intelligence to Drive Business Transformation: A Gartner Trend Insight Report

Overview

Key Findings

- There are several categories of conversational interfaces to satisfy increasing demand for such offerings. The landscape is hard to navigate because sales, marketing, media and customer assumptions are not aligned with the actual capabilities of the conversational platform.
- Many vendors still focus on curated learning, not machine learning, as the means to improve outcomes for users. The majority of end-user clients assume that artificial intelligence (AI) capabilities play a larger part in the overall platform.
- Most offerings in the market lack the sophistication needed for more-advanced implementations. Closing the gap will require substantial effort that is likely to not be possible for smaller vendors.

- Due to the similarities of components and capabilities in different conversational platforms, it is likely that the market segments for chatbot frameworks, VCAs, VPAs and VEAs will increasingly overlap and, in the future, merge into consolidated platforms.

Recommendations

Enterprise architecture (EA) and technology innovation leaders looking at how AI conversational platforms are evolving should:

- Choose vendors tactically through 2018 as the rapid pace of maturation in conversational technologies currently precludes making strategic choice.
- Consult our logical architecture map to understand the capabilities of conversational platforms and align vendor offerings with your needs, to avoid being stuck with a platform that does not fit your requirements.
- Plan to consolidate conversational technologies by 2019 into a conversational platform approach that can serve multiple use cases in the enterprise.
- Select only solutions or capabilities that have analytics, allowing you to learn about and improve conversational implementations.

Analysis

We are seeing the emergence of AI-based conversational platforms, which are driving the visibility and adoption of AI in the enterprise (see "Conversational AI to Shake Up Your Technical and Business Worlds" (https://www.gartner.com/document/code/315689?ref=grbody&refval=3621342)).

Vendors are taking different approaches toward satisfying user demand for conversational services. In the space of conversational platforms we find:

- Chatbot frameworks
- Virtual customer assistants (VCAs)
- Virtual personal assistants (VPAs)
- Virtual enterprise assistants (VEAs)

While there are large outward differences between these platforms, they also share many similarities in the underlying logical architecture and technological components. In fact, they share enough technology that Gartner predicts not only consolidation, but evolution in each individual segment (chatbot frameworks, VCAs, VPAs and VEAs) toward a unified market for conversational platforms within three to five years.

Customers looking at multiple use cases within the enterprise will benefit from a platform approach, which will also likely drive the convergence of these disparate market segments into unified platform offerings.

As the market for conversational AI platforms emerges, there will be a capabilities race. This research looks at the offerings of these four different segments that make up this market, and breaks down the common technological architecture and capabilities.

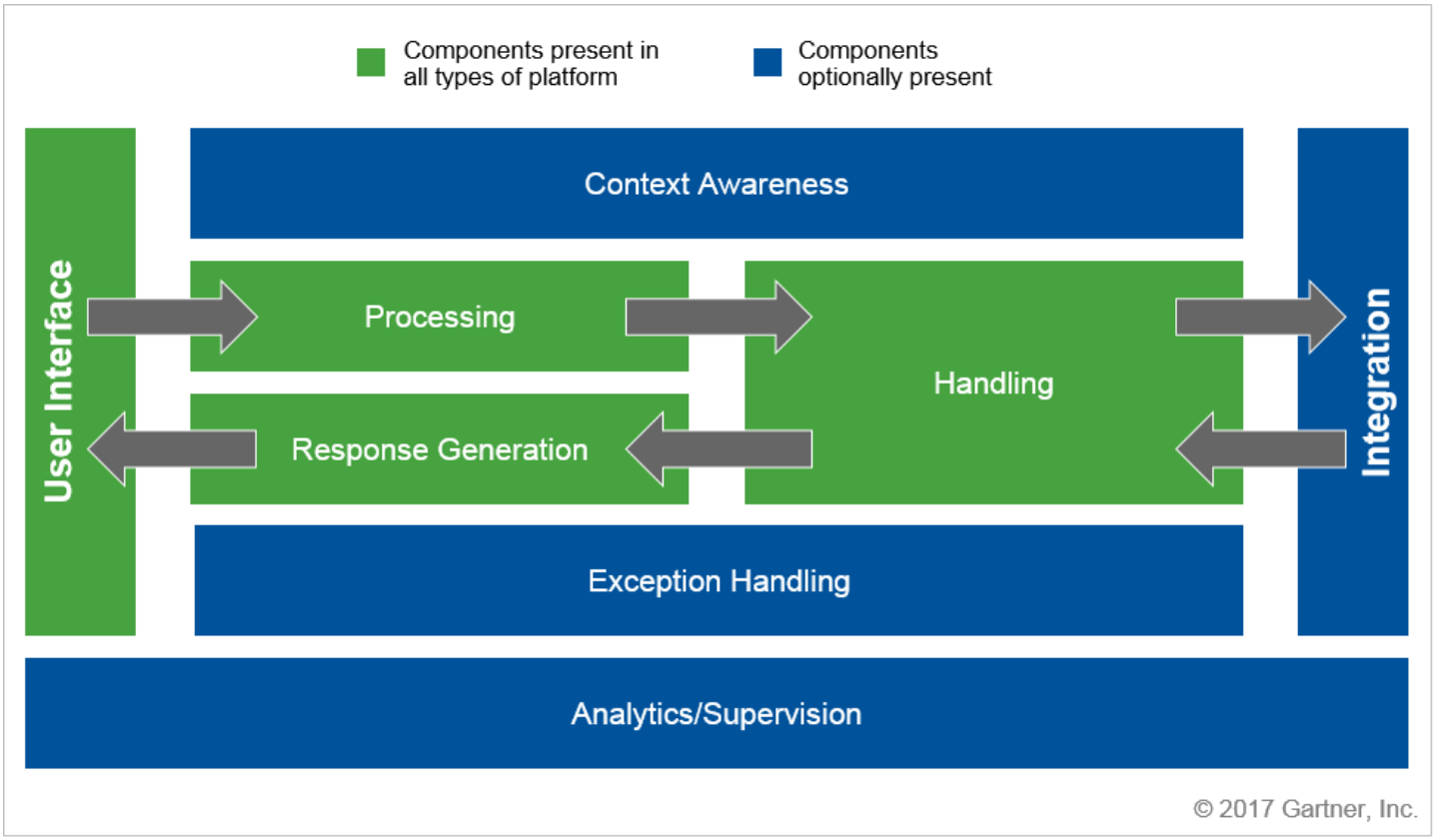
What we find is that the simplest architecture and capabilities are relatively easy to replicate. More-sophisticated architecture requires major research breakthroughs and potentially multimillion dollar research initiatives.

As we break down the capabilities, we are noting those currently present in almost all offerings, as well as those present in very few. We are also noting capabilities that are being researched and may not even be in the market currently, but that we believe will have to be present in the future.

Architecture

Figure 1 presents a representation of high-level architecture.

Figure 1. Logical High-Level Architecture



Source: Gartner (March 2017)

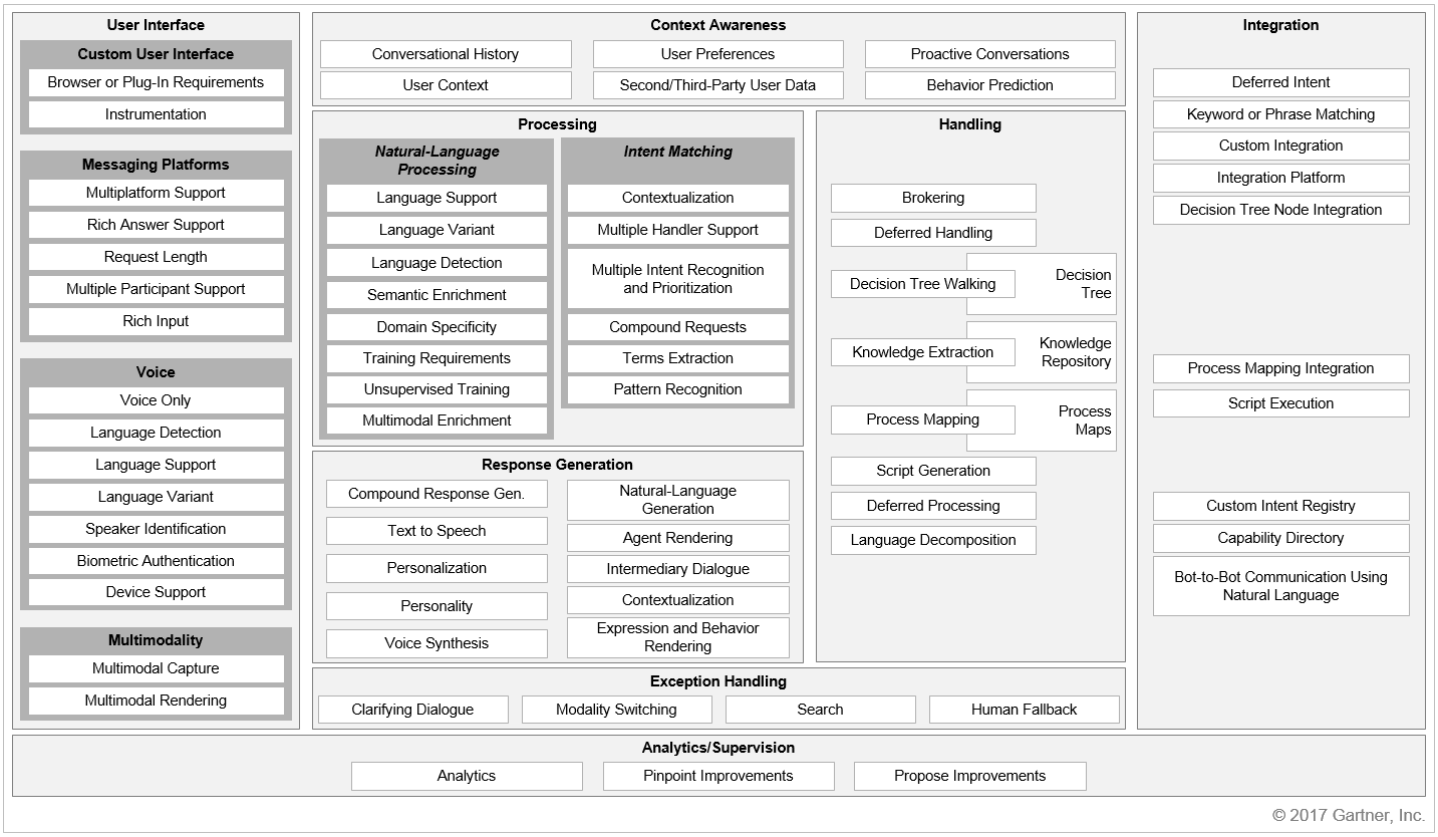
The high-level architecture of all types of platform is identical: A user interface is used to capture either voice or chat input, that input is processed and passed along to be handled, then a response is generated (in many cases just passed along from handling) to the user interface.

The simple offerings we see in the chatbot framework space consist simply of a messaging platform connector, a natural-language processing (NLP) engine, mapping of intent and a decision tree that gives

responses. Vendors that offer only this, however, will not keep pace with the future requirements of conversational platforms.

When we expand the architecture to cover all capabilities we have so far cataloged, we see greater complexity and sophistication (see Figure 2).

Figure 2. Expanded Logical High-Level Architecture — Capabilities and Components



Source: Gartner (March 2017)

Understanding both the high-level architecture and its capabilities is essential to be able to evaluate and compare vendors. This becomes especially important as the market is new and volatile.

We are already seeing certain vendors having a presence in multiple segments or breaking out of the core, mainly fueled by an increase in capabilities. Examples include:

- *Microsoft* — Present in all categories, with different overlapping offerings
- *IBM* — Offers both products and custom-made offerings more in line with a general-purpose platform
- *Google and Amazon* — Seeing synergies between their VPA and chatbot platform offerings

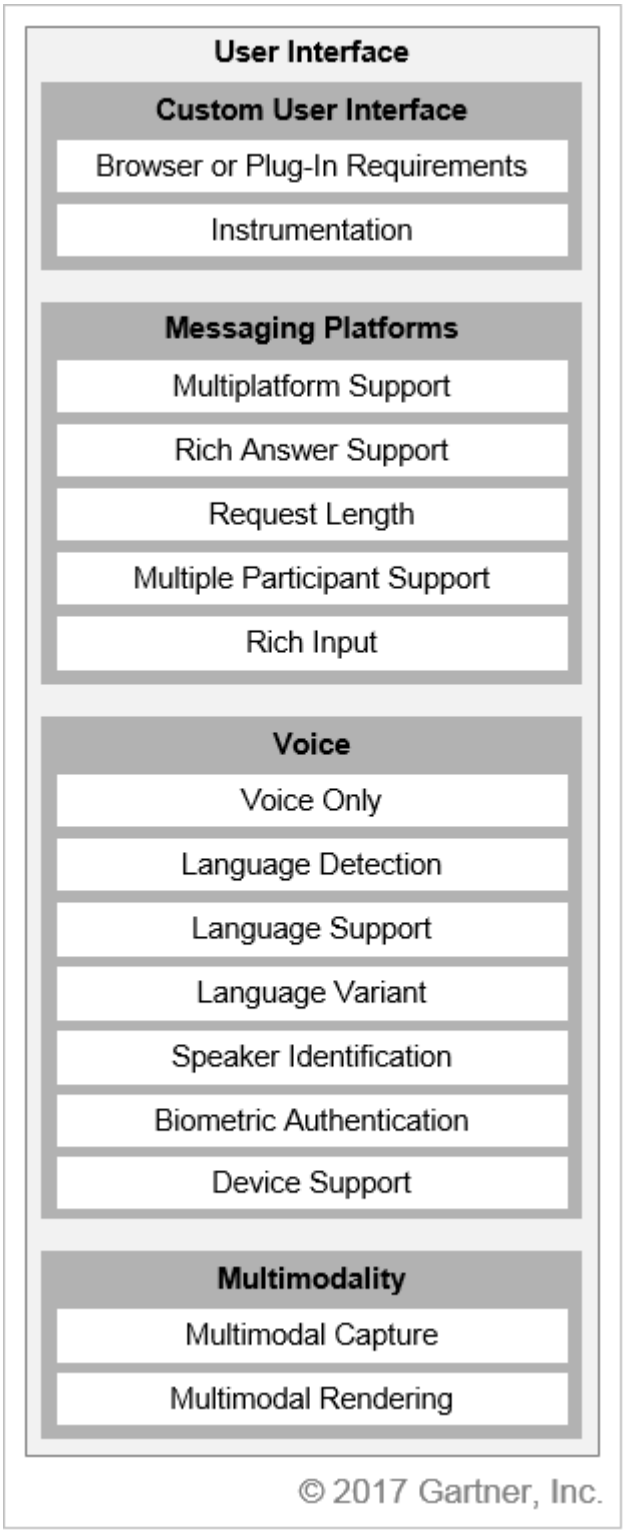
Capabilities

User Interface

There are several capabilities contained within the user interface (UI) that are highly dependent on modality, with different considerations needed for chat- than for voice-enabled interfaces (see Figure 3). In many

chatbot frameworks and VCA implementations, the UI is not part of the platform — the conversational platform simply acts as a user on the communication and messaging platforms that human users already use to communicate between themselves (examples include Facebook Messenger, WeChat and Kik). In the VPA and VEA space, tight control of the chat user interface is more common.

Figure 3. User Interface Capabilities



Source: Gartner (March 2017)

Chat

Chat is an interface that captures typed dialogue between two or more participants. All conversational platforms support one or more types of chat. Some leverage other chat platforms, some provide their own, and some do both.

Custom UI

- **Browser or plug-in requirements.** Sometimes chat implementations will use browser capabilities such as WebRTC, which has varying support in browsers, or browser plug-ins that may not be supported or may create significant barriers for users. Make sure to evaluate the support for web browsers.
- **Instrumentation.** Whether web-based chat or a module that can be plugged into mobile apps, the ease of instrumentation is an important factor to consider. While some solutions might only require a single JavaScript tag, others might need deeper and more time-consuming instrumentations.

In addition to the above, many other considerations for messaging platform integration also apply. Think of the vendor's own chat interface as a separate messaging platform for the purposes of evaluation.

Keep in mind that some vendors controlling their own chat implementation claim that they have a conversational interface, yet all the user is doing after the initial request is clicking on predefined question options. While this may suffice for certain use cases, it is not scalable to others.

Integrations With Messaging Platforms

The rise of messaging platforms such as Facebook Messenger, WeChat and Kik coincides with the rising hype surrounding conversational platforms, especially chatbots. But messaging platforms also include older technology such as email and SMS as potential channels that a conversational platform can use.

- **Multiplatform support.** What messaging platforms does the conversational platform support? Although some vendors claim to support multiple platforms, you may need to develop or tailor a chatbot for each one. Don't forget SMS and email as potential platforms as well.
- **Rich input.** Usually chat supports only text, but sending pictures or other nontextual information would be a natural evolution.
- **Rich answer support.** While all messaging platforms support simple text-based responses, there is also varying support for richer responses. Examples include WeChat's capability to serve mini applications as a reply, or iOS Messenger's capability to serve limited interaction elements such as graphics and buttons. Using the unique capabilities of the messaging platform — which users might expect — needs to be balanced against the need to support multiple platforms.
- **Request length.** Consider the potential length of requests with chat messaging platforms — especially if you include email as a way to communicate with the conversational platform. If the platform's NLP engine is optimized for short requests and single intents, applying that platform to email, which lends itself to long descriptions and multiple intents, would be a bad match. That market offerings currently have limited support for email is presumably because the complexity in handling the request increases exponentially as the length of the input increases.

- **Multiple participant support.** Messaging platforms support group chat. Make sure that this can be accurately detected and accounted for, as it might lead to novel use cases. A particularly interesting scenario for conversational platform use is to have a conversation between two humans, who after a while invite a chatbot into their conversation. The chatbot then has the whole conversation up to that point as context to further interactions.

Voice

Conversational platforms able to handle voice input offer varying degrees of capabilities.

- **Voice only.** The ability to handle all interactions using voice without ever falling back on presenting rich output such as search result lists, pictures and maps that require a screen.
- **Language detection.** The ability to detect what language is spoken and automatically switch to an engine supporting that language. In many cases language has to be explicitly set by either the user or in configuration of the platform.
- **Language support.** The ability to handle interactions in particular languages. Support for languages needs to be evaluated on quality, with variants, dialects, slang and accents all capable of confusing the speech-to-text engine.
- **Language variant.** The ability to handle interactions in different variants of the same language — for example, French and Canadian French, Norwegian Bokmål and Norwegian Nynorsk, or formal and casual Japanese.
- **Speaker identification.** The ability to identify different speakers. This is especially important in multiple-user scenarios where contextual data is being used for language processing, intent handling and/or response generation.
- **Biometric authentication.** The ability to not only identify a user based on voice patterns, but also authorize it. This is important if the conversational platform has requirements for security or privacy, or if risk for possible fraud needs to be reduced. Even more-sophisticated solutions with different levels of access for different users can also be envisioned.
- **Voice pattern analysis and enrichment.** A lot of information and emotion is conveyed outside of the words we use, by varying our tone, pace, pitch and other factors. By doing voice pattern analysis and enriching the generated text with additional meta information, some voice user interfaces are attempting to account for this additional information. In these cases, the fact that the platform is capable of doing this doesn't mean that it automatically accounts for this information in the generation of output — it merely adds the possibility if you are prepared to take on additional complexity.
- **Device support.** Voice recognition works very well when all factors of the environment and hardware can be controlled. But unknown variable quality can become part of the chain, such as environmental noise, not having control of the microphone or sampling rate from a device, or attempting to do far-field voice control without appropriate hardware. When this happens, the quality of voice recognition degrades.

Knowing the scenarios of use and the degree of control over hardware is important to determine if a voice interface can be a viable option.

Keep in mind that vendors often use white-labeled components for voice recognition, with varying degrees of customization. The number of good voice recognition engines in the world is far fewer than the number of voice-enabled conversational platforms.

Multimodality

While conversational platforms will naturally focus primarily on dialogue in the form of chat and voice, capturing other sensory data has the potential to improve accuracy and the quality of the experience.

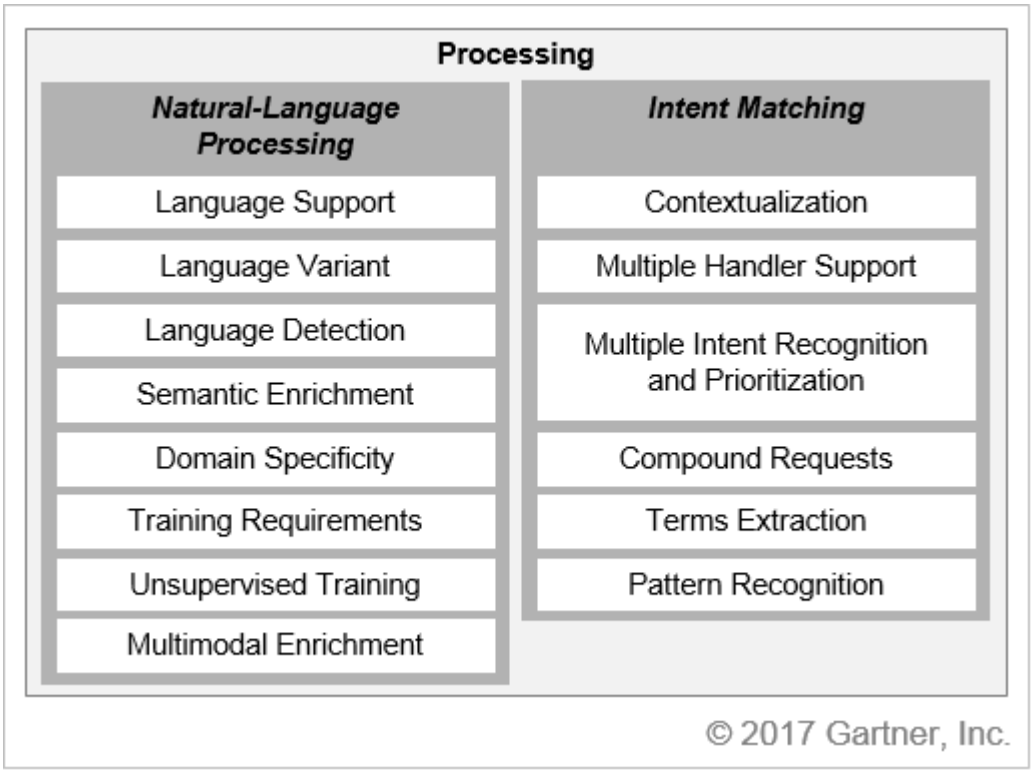
- **Multimodal capture.** It's highly likely that voice recognition will be augmented with data from video taken by the camera and other sensors. [1] (#dv_1_major_companies) We are starting to see support in conversational platforms for gesture recognition, facial expression recognition, face recognition and biometric authentication. In the case of stand-alone devices such as home speakers or in-car systems, additional specialist sensors might be built in to improve accuracy and the overall experience. Support for these additional signal sources as well as for processing them needs to be mirrored in the architecture.
- **Multimodal rendering.** This is the capability to not only reply in chat or voice, but render other means that also add to the exchange of information. This could be simple body language rendered on a virtual agent or expressions by a physical robotic assistant.

Processing

Behind the UI, a conversational platform needs to process input before it's able to handle the response. This part is required of any conversational platform. In many cases it's also the only part that contains machine learning, even though the product is labeled as an AI product. Finally, the processing is in many cases done by a white-labeled — perhaps customized — engine or API from another vendor.

Processing can be split into two steps: NLP and intent matching (see Figure 4).

Figure 4. Processing Capabilities



Source: Gartner (March 2017)

Natural-Language Processing

The NLP step is where the input text, along with additional information from the UI and contextual awareness cues, is processed in order for the conversational platform to understand.

In the case of voice-enabled UIs, language and language variant support might be tightly integrated; but in many instances, voice support simply converts the input from speech to text, and NLP has to be performed on the textual output from the speech recognition.

Note: For the purposes of this note, NLP and intent matching are two steps, but in reality they are tightly integrated and difficult to separate in actual implementations.

Typical capabilities of NLP include:

- **Language support.** What languages are supported. Quality might vary widely between implementations.
- **Language variant.** Different users have different styles when writing in chat interfaces. For some languages, different variants may need to be supported — for example, American and British English, Norwegian Bokmål and Norwegian Nynorsk, or French and Canadian French.
- **Language detection.** Does the language need to be explicitly chosen by the user, or is the language automatically detected from the text? This is especially important when several languages overlap the same geography, such as in parts of Europe or Asia.

- **Semantic enrichment.** In the NLP step, text is typically enriched with semantics based on the internal knowledge base of terms and expressions (for example, tagging names, companies, actions and so on that are mentioned in the text). NLP engines show a great variety of sophistication in this step, and it also varies a great deal even among individually supported languages in the implementation.

- **Domain specificity.** There are several layers of specialization possible in the NLP engine. A typical general-purpose vocabulary, out of the box, will most likely be ill-suited for most implementations. In most projects, a lot of time is spent training the NLP engine to an acceptable level of performance.

  Domain specificity can take the form of:

  - *Industry specificity* — Having a vocabulary tailored to understand banking, insurance or travel
  - *Purpose specificity* — Having vocabularies tailored to understand in the context of being used for IT service desk or calendar scheduling
  - *Customizable specificity* — Being able to manually configure synonyms, terms and phrases
  - *Trained specificity* — The vocabulary that's the final result of training the NLP engine with training data

  A possible variant on domain specificity is a broker pattern, where an engine specialized on simply detecting the domain will qualify the input to one or more underlying domain-specific solutions.

  Consider if the vendor is giving access to general-purpose vocabularies or vocabularies with a level of specificity. Also consider the level of training that will be necessary to get acceptable performance. As a general rule, training needs are highly dependent on the scope of the implementation.

- **Training requirements.** Even if an implementation has good performance out of the box on a specific language, it might need a lot of additional training data in order to achieve the expected results. In some instances, that training data might not even exist, and thus a customer might be stuck with a poor-performing implementation until enough training data can be gathered.

- **Unsupervised learning.** Does the NLP engine have to be trained with training data only, or will it be able to adjust the model after deployment based on the result of the conversation with the user? In most cases the unsupervised learning will still be an offline activity, but some chatbot implementations have tried "real-time" unsupervised learning. Be aware of potential "attacks" against a continuous unsupervised-learning NLP, since it can be taught to respond wrongly with a coordinated effort.

- **Multimodal enrichment.** This is taking data collected by other sensors to enrich the processing and results. As an example, facial expressions may enrich a statement with information that makes it more likely to be interpreted a certain way, such as ironically.

Intent Matching

Intent matching is where the processed input is matched to the appropriate handler of the request. This usually uses machine learning.

Note: For the purposes of this note, NLP and intent matching are two steps, but in reality they are tightly integrated and difficult to separate in actual implementations.

Several capabilities are possible:

- **Contextualization.** The ability to make current contextual cues part of intent matching. Simple architectures will not consider context when matching.
- **Multiple handler support.** The simplest implementations have just one handler, which is usually a decision tree. The simplest intent matching consists of just matching input to the appropriate point in the decision tree. Additional complexity is introduced if it's possible to match to multiple types of handler.
- **Multiple intent recognition and prioritization.** The simplest implementations support just one intent per input, which means that requests with multiple intents need to be explicitly accounted for. If supporting longer-form formats, such as email, the ability to handle multiple intents become important. Take this simple example:

  "I want to order a pizza, but I'm unsure if you deliver to my address?"

  The answer to this would be greatly improved if the intent matching is able to detect both "want to order" and "check delivery coverage." Being able to prioritize the two intents is important to pass along to the handler.
- **Compound requests.** Similar to multiple intents, compound requests are multiple in number. But instead of affecting each other, they are completely separate. An example:

  "I want to order a pizza, and I also want a Coca-Cola to drink as well as the movie on offer."
- **Terms extraction.** A user might have the same intent, but the request might contain additional information for the intent handler. An example:

  "I'd like to order a pizza with marinated chicken, four cheese and tomatoes."

  The intent is "order pizza," but a list of ingredients is also given to the handler.
- **Pattern recognition.** This is more advanced than terms extraction, in that the intent matching is capable of matching intent with unknown terms in it by looking for particular patterns in the request. An example:

  "Play a song by Prince on Spotify."

  If doing terms extraction, this would need knowledge of all potential artists; but if doing pattern recognition, the nature of the pattern would indicate the intent rather than the specific terms used. More-advanced implementations of pattern recognition would take context into consideration as well.

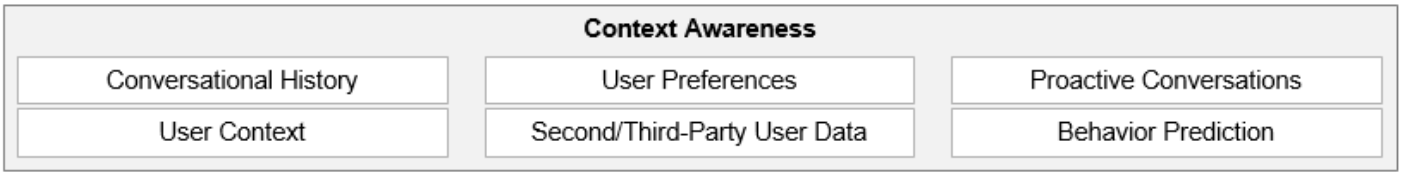
Since intent matching is commonly based on machine learning, take special care when evaluating the cost of maintenance and governance. This can easily rise as complexity increases. As an example, in the case of an engine with multiple handler support, what would happen if the decision tree were updated with an

answer that was handled by knowledge extraction? Will the machine-learning model in this instance be capable of updating based on a priority of handlers, or will it have to be retrained? As a rule of thumb, the more advanced the intent matching capabilities used, the more training data is required to make it work at an acceptable performance level.

Contextual Awareness

Being aware of the context of use is of vital importance to a conversational platform's ability to successfully match intent (see Figure 5).

Figure 5. Contextual Awareness Capabilities



Source: Gartner (March 2017)

- **Conversational history.** The ability to learn from previous conversations and reuse that information in future conversations becomes increasingly important as frequency of use increases (like in VPA, VEA and some chatbot use cases). It is also a prerequisite for a VCA that has the ambition to solve more-complex requests that span several interactions or offer hyper-personal service.
- **User context.** This is the ability to take into account outside contextual cues. Examples include on what page a website chat is used, previous pages visited or geographical location gauged from the mobile phone GPS.
- **User preferences.** These are especially important for implementations that see frequent use. There might be a requirement for users to access preferences and turn off certain capabilities that they are not comfortable with or want enabled. This can take the form of settings screens in an app, or even the ability to set preferences through the dialogue (e.g., *"I prefer that you call me Bob"*).
- **Second/third-party user data.** This is the ability to leverage outside data about the user. This can take the form of CRM data or information from a public profile on Facebook, for example.
- **Behavior prediction.** This is the ability to predict the behavior of a user based on past interactions or past interactions with others. By itself, this is not terribly useful except to potentially skip unnecessary steps in dialogue; but combined with proactive conversations (below), this has the potential to be very useful.
- **Proactive conversations.** A very exciting development is the capability of conversational platforms to initiate conversations instead of only responding to users. To be able to do this, the platform would need integrations with some kind of event processing and either explicit or implicit rules for triggering starting a conversation.

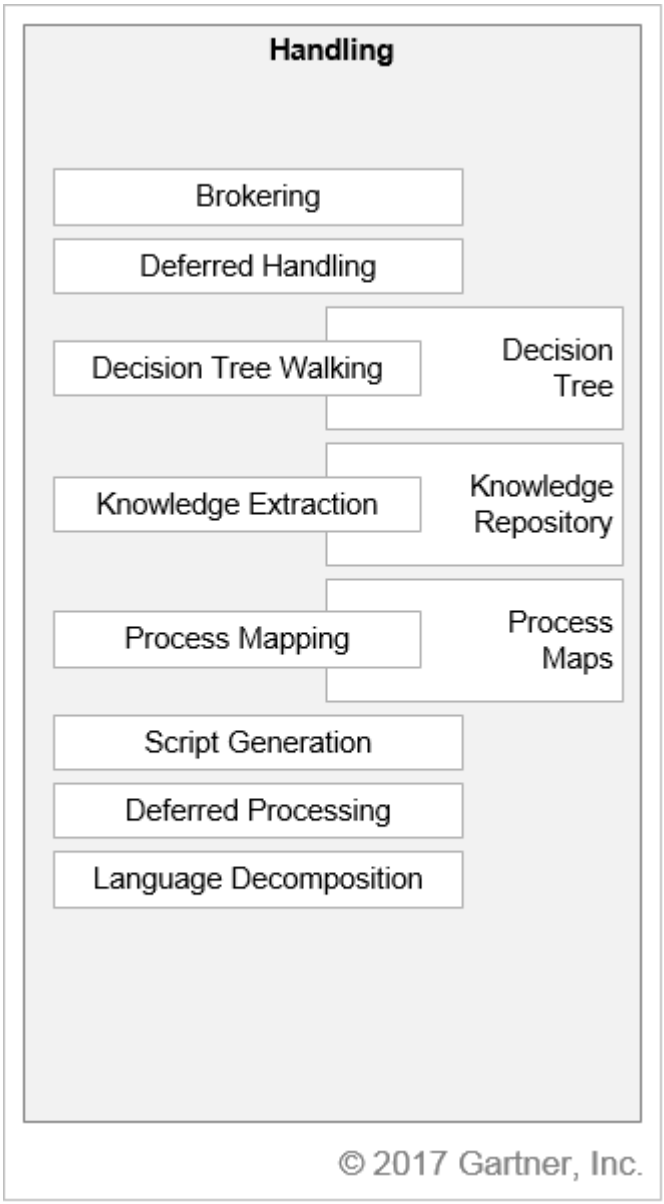
The line between helpful and irritating is, however, very hard to navigate. This kind of functionality requires not only careful considerations of the negative consequences, but extensive testing and validation.

Handling

Handling requests using decision trees is by far the most common handling method in conversational platforms. This often comes as a surprise to customers, since vendors like to attach the labels "machine learning" or "AI" to their offerings. In the case of a decision tree, there is no machine learning involved, except in how requests are mapped from natural language to the intent that signals the entry point into the scripted dialogue.

This is a field where we see a lot of potential for innovation. We fully expect to see new ways of handling in the future — especially considering other capabilities such as multiple and compound intents that will quickly grow decision trees to unmanageable sizes, necessitating the need for other handling methods (see Figure 6).

Figure 6. Handling Capabilities



Source: Gartner (March 2017)

- **Brokering.** In the cases where a platform has multiple ways to handle intents, a mechanism to broker intents among the different handlers is necessary.
- **Deferred handling.** The easiest handling is to just pass the processed request along to a custom-developed service. For specific-purpose chatbots, this might be all that is needed, but other kinds of handling are required for the majority of implementations. Even if a conversational platform has sophisticated handling capabilities, the option to defer handling of certain requests can still be a way to enhance the experience and enable truly differentiating experiences.
- **Decision tree walking.** By far the most common way of handling requests, this consists of a dialogue tree where each node is matched to an intent and contains potential subnodes to keep the conversation going. For each node there might be an option to respond with a standardized response or pass along to an API. Consider:

- How the tree is created and maintained (visual tools, in code, configuration files and even needing professional services are all possible)
- How sophisticated the tree can be
- How scalable the tree will be if it grows large
- How sophisticated the handling on each node can be — including the response

The main advantages are that decision trees are fully transparent and answers can be fully controlled, allowing for personality, humor and reflecting consistent brand values.

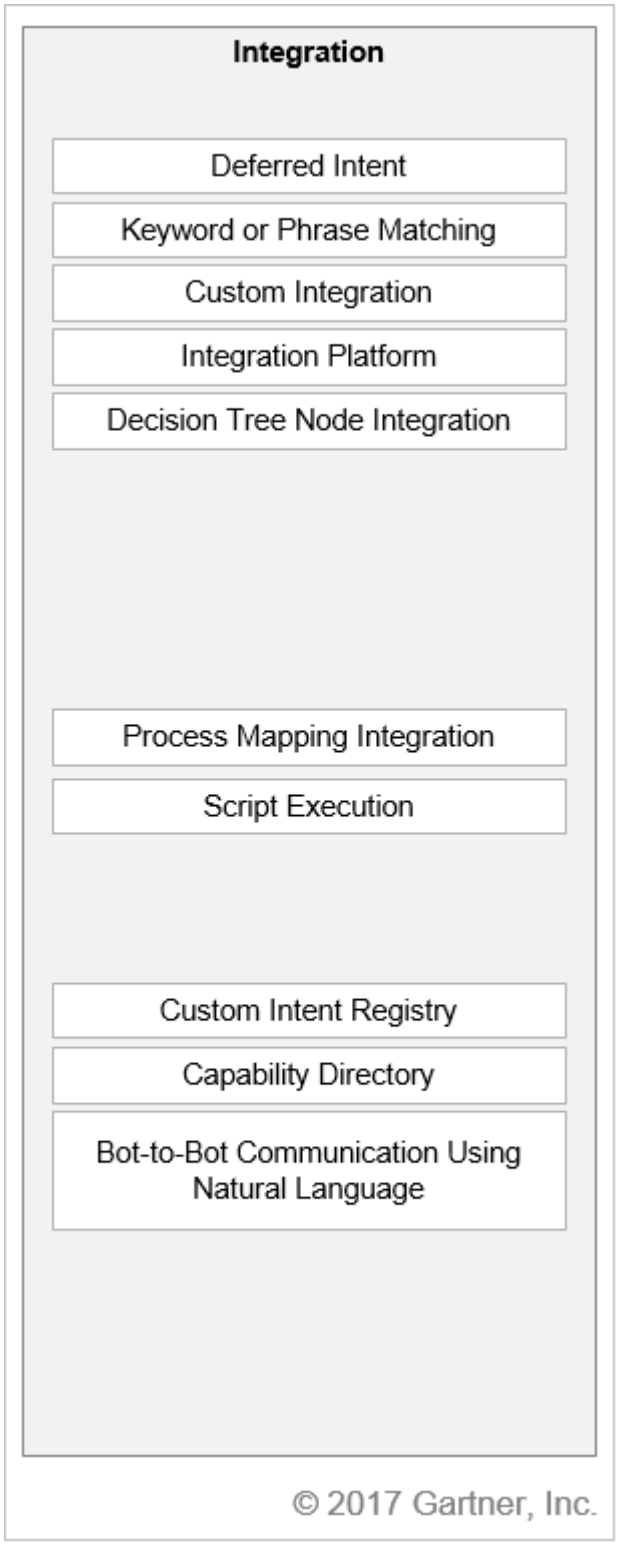
- **Knowledge extraction.** Also called fact extraction, this is the capability of turning a request into a query. The relevant information is extracted out of a large knowledge or content repository, and presented back to the user as an answer.
- **Process mapping.** This is the capability to map conversations into steps in business processes, focusing the conversational elements on what is "current state" and what information or action is needed to move to the next step in the process.
- **Script generation.** Also called query generation, this is the capability to translate output from the processing step into a more formalized scripting language that is executed in a script engine. The script engine could enable other handling mechanisms and direct integrations.
- **Deferred processing.** Instead of just deferring the handling, the whole unprocessed request is passed along to another conversational platform or bot implementation on the same platform. Together with language decomposition, this can enable the handling of complex general-purpose requests by multiple specific-purpose implementations.
- **Language decomposition.** This is the ability to decompose complex requests into simpler separate statements, which are then run as separate requests to different handlers. This would require the ability to compound answers in the response generation. For each type of handling supported, the amount of work involved and the toolsets to do that work should be evaluated.

Vendors that currently offer only deferred handling and/or decision tree walking will struggle to keep pace in this area. The research and development effort required to implement the other handling methods is exponentially larger, and many will not be able to cross that chasm. Considering this, the odds of future support for sophisticated handling favor the big software vendors.

Integration

While stand-alone chatbot implementations have a purpose, many use cases require integration with existing systems. There are several ways that this integration may happen (see Figure 7).

Figure 7. Integration Capabilities


Integration
Deferred Intent
Keyword or Phrase Matching
Custom Integration
Integration Platform
Decision Tree Node Integration
Process Mapping Integration
Script Execution
Custom Intent Registry
Capability Directory
Bot-to-Bot Communication Using Natural Language
© 2017 Gartner, Inc.

Source: Gartner (March 2017)

- **Deferred intent.** After processing and matching intent, the request is passed along to a system that has registered itself to handle that particular intent. It requires the implementation to control the conversation, but allows for interchangeable services to execute the requests. An example would be having either Uber, Lyft or the local taxi service register for a "get a car" intent. It requires the registering service to implement a known API to handle the deferred request.
- **Keyword or phrase matching.** Integration through registering a keyword or phrase that, when used by a user, triggers deferring of handling to the service. An example would be, *"Tell Spotify to play some Christmas music."*
- **Custom integration.** Custom integration simply means that any integration needs to be custom-coded for the implementation.
- **Integration platform.** In more-sophisticated platforms, a third-party or custom integration platform may be enabled to ease consumption of APIs and managing integrations. If there is a need to integrate many back-end systems, this might be a necessary capability.
- **Decision tree node integration.** The ability to specify a RESTful method call to be executed from a particular node in a decision tree. This allows simple integration from a SaaS-based conversational platform to available RESTful APIs.
- **Process integration.** Integration from an internal representation of the business processes in a process mapping handler to a business process management workflow engine that handles the workflow and

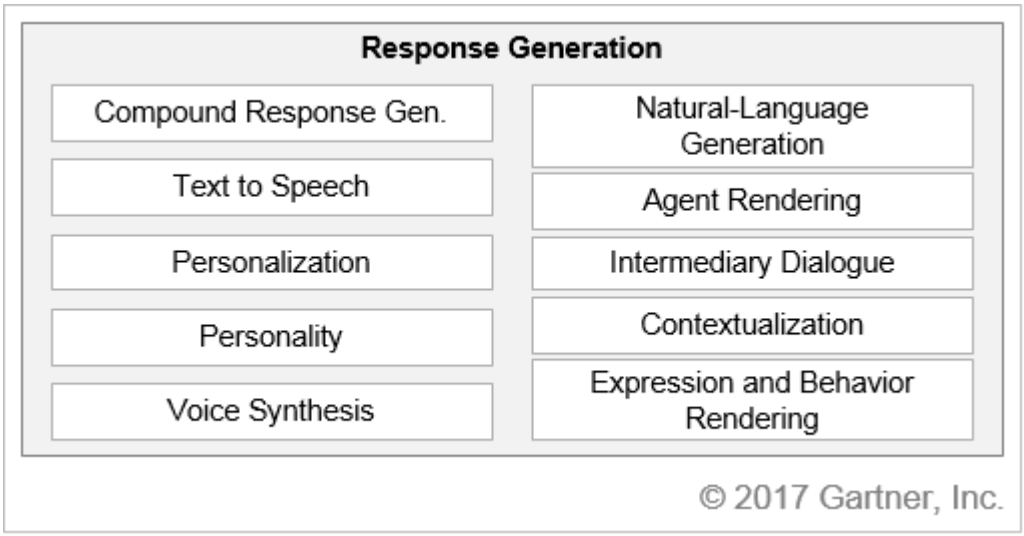
integrations.

- **Script execution.** In the case of script generation handling, any services wanting to integrate would need to implement an ability to execute the script. This allows for much deeper integrations than APIs, but also requires more implementation work on the part of the provider of the service.
- **Custom intent registry.** This is the ability to train your service in recognizing a particular intent (that can be unknown to the conversational platform), and register that intent along with the training to recognize it with the conversational platform.
- **Capability directory.** This is maintaining a directory of services that implement a particular API, one or more of which can be used when the capability is needed.
- **Bot-to-bot communication using natural language.** The ability of the conversational platform to turn requests into new requests that are then used to communicate with other bots (see "Maverick* Research: Machines Will Talk to Each Other in English" (https://www.gartner.com/document/code/291027?ref=grbody&refval=3621342)).

Response Generation

Anything more sophisticated than prescripted responses needs capabilities for doing response generation (see Figure 8) — at a minimum, natural-language generation (NLG).

Figure 8. Response Generation Capabilities



Source: Gartner (March 2017)

- **Natural-language generation.** The capability of generating natural-language responses based on structured data or other inputs. NLG is important if there are handling methods other than decision trees. Assess the amount of training needed to have the NLG produce good responses, and also the quality of the output from a readability perspective.

- **Compound response generation.** In cases where a platform supports multiple intents or compound requests and is able to handle these separately, there is a need for taking several answers and making them into one compound response.

- **Intermediary dialogue.** Sometimes processing, handling, getting data from integrations and generating an answer take time. If cloud processing of voice is necessary as well, responses may be delayed by seconds, creating awkward gaps of silence in the dialogue that degrade the experience. Intermediary dialogue (*"Hold on while I look that up"*) is similar to how a human would respond, and may help mitigate the latency problem.

  We expect this to evolve further, with some conversational platforms starting a response without knowing the actual result until later in that response (*"Looking at your accounts …* [lookup complete at this point]*, your balance is $X"*).

- **Contextualization.** The ability to use the contextual capabilities in the platform to tailor the response.

- **Personality.** The ability to add personality characteristics to automatically generated responses. Personality is of vital importance to the design of conversational experiences, and an accurate projection of brand values can make experiences better for users.

- **Personalization.** Beyond basic personality is the ability to not only tailor the personality to the current implementations, but take into account the writing or speaking style of the user, cultural cues and other factors, to truly personalize the response to an individual user.

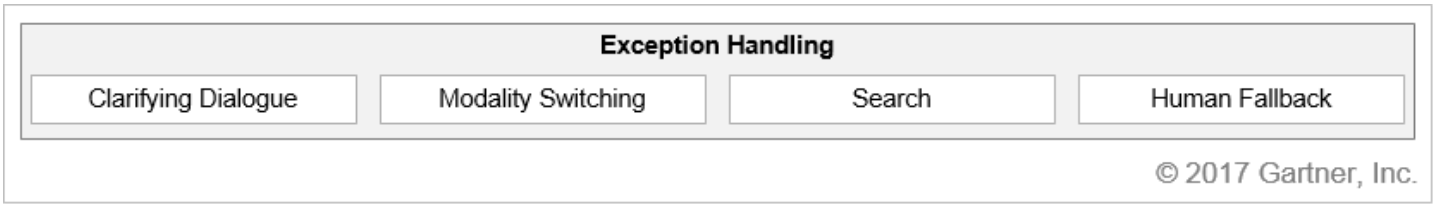
- **Text to speech.** In the case of voice interfaces, this is the actual voice generated by the platform. While usually slightly mechanical sounding, recent research developments have shown extremely accurate reproductions of human voice that are next to impossible to detect as machine voices.

- **Agent generation.** This is the generation of an agent, in the form of a humanlike body or face, or even a robot illustration. The generation includes facial expressions, lip syncing to voice, body postures and gestures. Consider this if there is a need for agents in the first place. Many vendors support this feature primarily as a way to brand their product.

Exception Handling

Exception handling, also called escalation, is the capability to route a request that is not understood, or poorly understood, to an alternative handling method. Considerations must be taken for what triggers an escalation — is it simply when an implementation is unable to answer? Or is when the predicted Net Promoter Score [2] (#dv_2_net_promoter) from the interaction falls below a certain threshold, necessitating a human takeover of the dialogue (see Figure 9)?

Figure 9. Exception Handling Capabilities



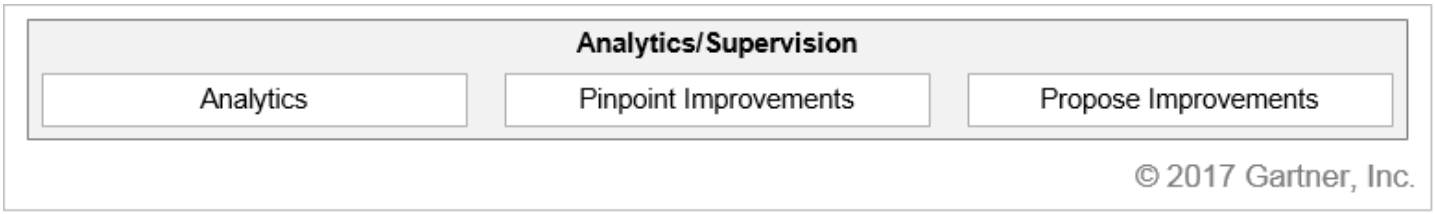
Source: Gartner (March 2017)

- **Clarifying dialogue.** When faced with multiple possible intents or an intent below the confidence threshold, this is the capability to ask questions that will improve the confidence threshold. This is a notoriously hard problem to actually solve, and may involve a lot of manual work to get correct.
- **Search.** The capability of passing along the request to a search engine that will present the user with search results. This will take the user out of the conversational paradigm, forcing a modality switch (below).
- **Modality switching.** The capability to, instead of answering in the conversation, direct the user to the appropriate service, app or website where the request can be fulfilled.
- **Human fallback.** The capability to pass along requests to a human who then takes over the conversation. More-sophisticated solutions not only pass to humans, but let humans pass back to the machine when the request has been clarified.

Analytics/Supervision

All solutions should have analytics. More-sophisticated platforms also give you the tools to turn that analytics into action and help you improve (see Figure 10).

Figure 10. Analytics/Supervision Capabilities



Source: Gartner (March 2017)

- **Analytics.** The capability to generate reports and look at the performance of the implementation. Considerations should be taken according to how the metrics are being used: Is the need just for reporting at regular intervals, or real-time monitoring?
- **Pinpoint improvements.** The capability to pinpoint potential areas of improvements — typically similar requests that are not being handled and similar answers given by human employees on fallback. While it pinpoints possible places to improve, it does not tell you what improvements to make.

- **Propose improvements.** The capability to monitor and propose new additions to the decision trees or other handlers. It often involves both machine learning to give proposals and human supervision to approve them.

Even when doing a simple proof of concept, analytics is of vital importance to learn and improve an implementation. Solutions that don't offer analytics are effectively running blind and should not be considered for any purpose.

What About End-to-End Machine Learning?

Some architectures exist where NLP, intent matching, handling and response generation are done in a single operation. These kinds of implementations typically use a deep neural network similar to that used for language translation, but instead of generating a translation, it generates a response in the same language. More-advanced implementations would also take the rest of the conversation as input, creating the possibility for long conversations.

There are several challenges in scaling this to the point where it will be a viable option for a conversational platform. There is a challenge with transparency. It takes great effort to understand why a particular answer is given to a particular question. There is also a problem in hooking into integration with other systems, especially because of the transparency challenge.

Gartner believes end-to-end machine learning has its uses, but will be of limited use on a platform for quite some time.

Using This Document Effectively

There are a number of ways to use this document effectively:

1. **Scoping.** Scope the kinds of capabilities you are looking for, in order to find a service or product that is a good fit.
2. **Evaluation.** Evaluate current capabilities and the roadmaps of vendors in order to determine future viability — will they scale with your needs in the future?
3. **Integration.** Determine future integration needs of existing systems and how they might fit in a conversational platform.
4. **General knowledge.** A tremendous amount of discovery is embedded in this logical architecture. It allows you to have conversations on an elevated level of sophistication with vendors, developers and designers.

Evidence

[1] Major companies, especially in the VPA market category, have done several acquisitions of startup companies that were working on gesture and emotion detection from regular webcam footage.

[2] Net Promoter Score is explained here: https://www.netpromoter.com/know/ (https://www.netpromoter.com/know/) .

Recommended by the Authors

Conversational AI to Shake Up Your Technical and Business Worlds (https://www.gartner.com/document/code/315689?ref=ggrec&refval=3621342)

Market Guide for Virtual Customer Assistants (https://www.gartner.com/document/code/291810?ref=ggrec&refval=3621342)

Market Insight: How to Collaborate and Compete in the Emerging VPA, VCA, VEA and Chatbot Ecosystems (https://www.gartner.com/document/code/319851?ref=ggrec&refval=3621342)

Seven Decision Points for Success With Virtual Customer Assistants (https://www.gartner.com/document/code/299432?ref=ggrec&refval=3621342)

Market Insight: Conversational Commerce — Hype or Reality? (https://www.gartner.com/document/code/311257?ref=ggrec&refval=3621342)

Market Insight: Disruptive Macro Trends for 2025 Personal Tech Market — Artificial Intelligence — Me, Myselves and AI (https://www.gartner.com/document/code/314504?ref=ggrec&refval=3621342)

Maverick* Research: Machines Will Talk to Each Other in English (https://www.gartner.com/document/code/291027?ref=ggrec&refval=3621342)

Maverick* Research: The Demotion of the Smartphone (https://www.gartner.com/document/code/315814?ref=ggrec&refval=3621342)

Recommended For You

Craft an Artificial Intelligence Strategy: A Gartner Trend Insight Report (https://www.gartner.com/document/3847266?ref=ddrec&refval=3621342)

Predicts 2018: Artificial Intelligence (https://www.gartner.com/document/3827163?ref=ddrec&refval=3621342)

Questions to Ask Vendors That Say They Have 'Artificial Intelligence' (https://www.gartner.com/document/3779264?ref=ddrec&refval=3621342)

The CIO's Journey to Artificial Intelligence: Learn Then Leap (https://www.gartner.com/document/3787067?ref=ddrec&refval=3621342)

Artificial Intelligence Primer for 2018 (https://www.gartner.com/document/3845465?ref=ddrec&refval=3621342)