Strategic Machines, Inc.

Agreement with Tresean Adam

June 1, 2018

For professional services:

Slack bot – Turing Machine

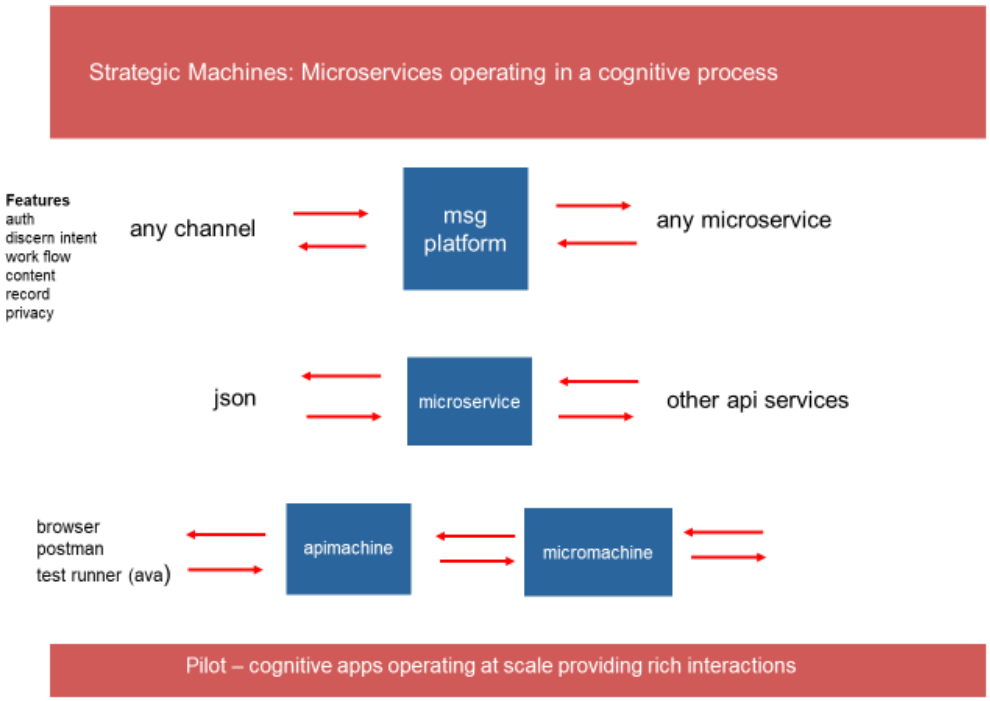
Backgrounder on the vision of Strategic Machines

The novel idea of Strategic Machines is that cognitive applications can be delivered in a short cycle at a lower economic price point by leveraging microservices. The term 'cognitive' has become synonymous with 'artificial intelligence', and as a result, many static or highly predictable tasks that can be handled through pure functions are being deployed on specialized ai platforms. As a result, complexity and costs mount as companies attempt to create 'conversational apps' using highly specialized languages and complex platforms. Organizations are reluctant to adopt. Opportunity is lost.

Strategic Machines is dedicated to the delivery of winsome Brand interactions through standard, open source technologies. We consume ai services where needed but build interactions in a manner that makes it easy to configure, test and maintain.

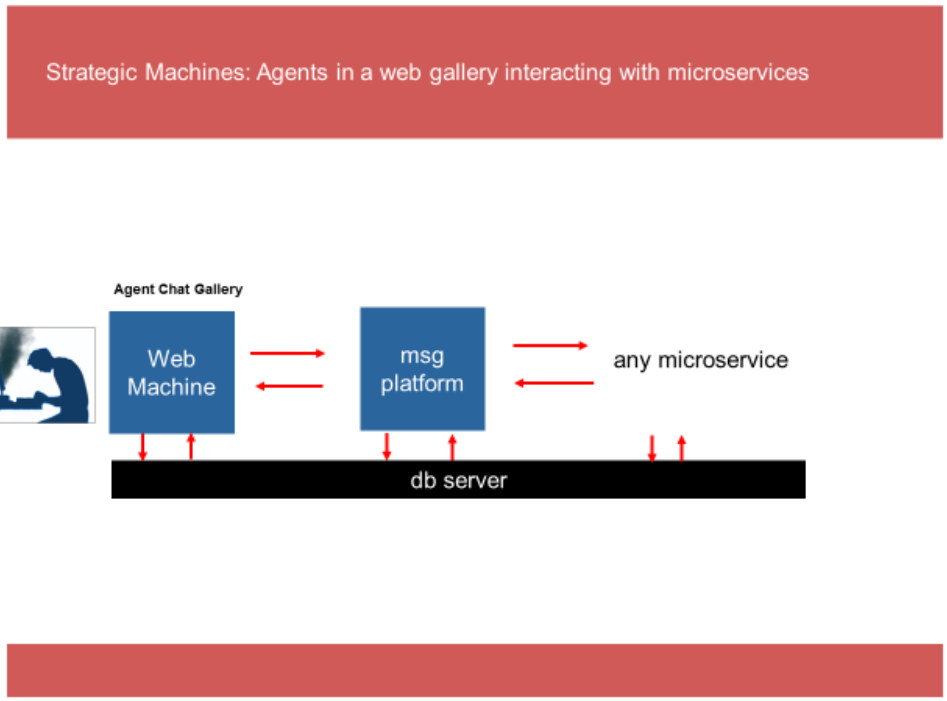
By engaging customers through any channel with smart virtual agents, we help our clients gain a competitive edge in their respective markets. Microservices is at the heart of the architecture.

As additional context note the following diagram



1. The ‘msg platform’ is designed to handle a text message from any channel (slack, sms, fb, telegram, viber etc.) The text message is authenticated, analyzed and passed to a microservice for processing as part of a uniform data object, along with a rich set of additional information that can be used by the microservice to compose responses.
2. The microservices receives a json data object, and processes that data object by initializing a constructor. The constructor (published as an npm package in the future) provides a set of functions for interrogating the value of properties on the data object and returns an array of replies to the messaging platform
3. The apimachine and micromachine are 2 simple servers for use by developers in testing new microservices (not the scope of this workorder – just provided here for information purposes)

The Web Machine platform provides a host of infrastructure services to register networks, join communities and configure agents. As illustrated in this diagram, an agent gallery will enable available bots to chat with a user – by tying into backend microservices through the messaging platform



Workorder:

This piece of work is our first integration with the Slack messaging platform. The objective is to build a slack app that can successfully connect with the Strategic Machine messaging platform, sending and receiving a transaction, and therefore facilitating a conversation. The ultimate functionality for this bot will reside in a microservice (outside of the scope of this workorder).

1. Build a slack bot ('Turing Machine') and deploy the bot to a new Strategic Machine Account which will serve as the landing platform for customers. The bot will eventually be 'activated' by a microservice – and will be composed to interact with the customer about the capability of the Strategic Machine platform. In other words, this is a 'sales bot' – and will help to convince a customer to sign up for services. The scope of this bot however is only the front end (i.e. the UI)
2. This piece of work involves creating the bot – and integrating it to the apimachine so that it can fire off an api and trigger a 'test microservice (resident on the micromachine). The bot needs to take a request from the user, trigger the api passing the request (i.e. text string) and receive a response form the apimachine and ender a response in the slack channel
3. As part of the scope of this work, documentation on steps to create a bot will be included, as well as the docs on the bot itself
4. Some potentially useful tutorials and repos to explore include

   https://api.slack.com/tutorials

   https://www.npmjs.com/package/slackbots

   https://github.com/gtracy/slack-stuart

This work order is exploratory in nature – so it is designed to provide you with time to research and build Strategic Machine's first prototype slack bot, but to also think through how we can 'systematize' the approach and model for building an array of slack bots for various purposes in the future.

As an approach to this workorder, I recommend you use the tutorials to earn the mechanics of slack bot building, but then lay out the design for the bot to 'message platform' workflow (using apimachine as your proxy platform). Once that design workflow is completed – lets have a conversation to discuss and 'test the design' before you begin constructing your prototype.

In addition to the set of requirements outlined above

- Demonstrate disciplined and well-organized coding techniques for all modules delivered. Ensure that code is well structured and commented
- Tests are normally included as part of a deliverable, since in this work order it is not clear how automated tests can be applied against the slack platform, the test requirement is suspended
- Code will be delivered through pull requests on the 'strategicmarket' github site consistent with the codex. Obviously, any code for the slack bot will be delivered in the registry on the new Strategic Machines Account
- Expected delivery date of final deliverable is targeted for no later than June 16, 2018.
- You agree that this is work for hire, work will be done in a quality manner, all deliverables are owned by Strategic Machines, Inc, and the work or concepts will be treated confidentially and not be disclosed to another party without express written permission of Strategic Machines
- This work order can be terminated by either party at anytime

Payment:

$240.00

upon delivery of this tested application. Any specific instructions for the operation or configuration of the web page should be delivered as part of README file inside the directory for this app.

Your signature on the line below indicates your agreement to the scope and terms of this work order.

Keep 1 copy of this signed workorder for your records and return 1 copy to Strategic Machines

---

Agreed to:

Date: