Strategic Machines, Inc.

Agreement with Evgheni Pascalov
May 29, 2018

For professional services:

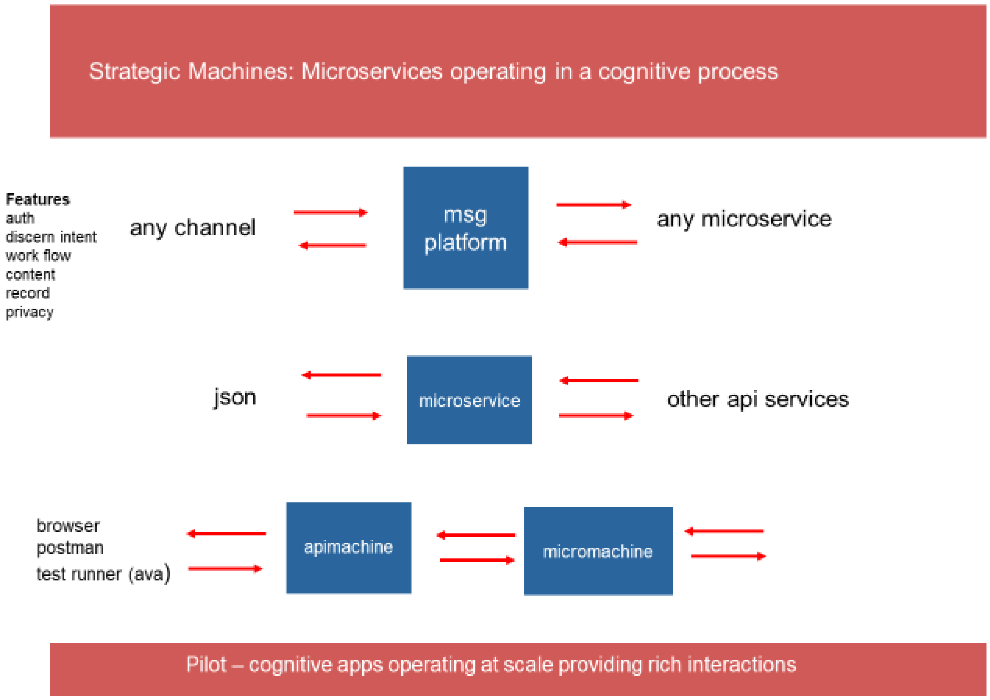
Agent Chat Gallery

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 to specialized ai platforms. As a result, complexity and costs mount. 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 were 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

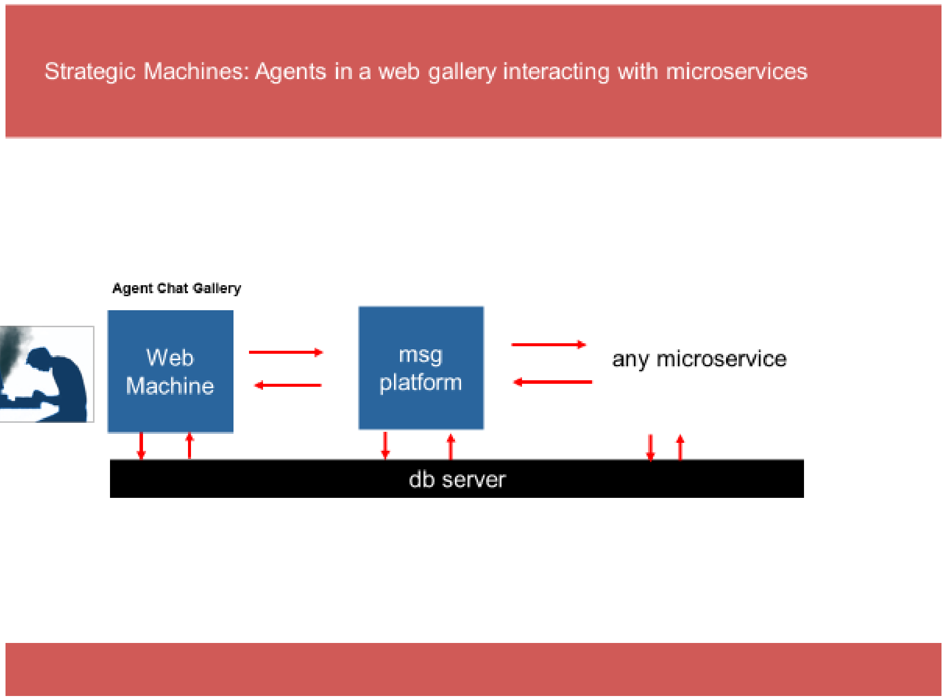


1. The 'msg platform' is designed to handle a text message from any channel (slack, sms, fb, telegram, viber etc.) The text message is authenticated, analyzed and passed to a microservice

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.

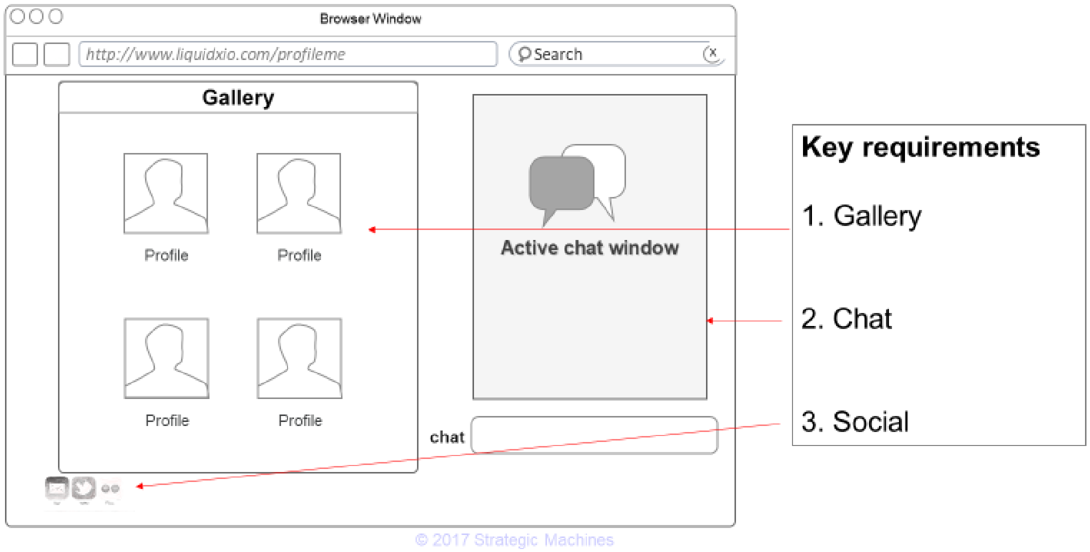
2. The microservices receives a json data object, and processes that data object by initializing a constructor. The constructor (published as an npm package in the future) provides a set of functions for interrogating the value of properties on the data object and returns an array of replies to the messaging 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 scope of this work order is capture in the following 2 diagrams



Strategic Machines: Agents in a web gallery interacting with microservices

1. The agent gallery web page will be integrated with the newly refactored webmachine platform. Once integrated, issue a pull request so that the platform can be redeployed to the cloud, demonstrating that it is operational
2. The agent gallery will be refactored to communicate with the messaging platform via an api. A test version of the platform will be available in the cloud, and accessible through an api. (contact Pat when you’re ready for information on the messaging platform api)
3. Every agent in the gallery can be ‘activated’ for chat. The chat session drives the set of microservices which comprise the ‘cognitive ability’ of the agent to respond based on some topic. The only requirement for the gallery chat is to receive the JSON object from the platform – and display the reply (returned as an array)



The reactjs web page has 3 components.

1. The gallery, which would include a 'profile' of the agent based on available information in the agent collection
2. The active chat window for what every agent was selected (the chat window should provide a visual cue (name, image) of the agent with whom the user is chatting
3. A set of social buttons (these do not have to be activated but do need to be usable and ready for configuration)
    a. Slack – if the user wants to sign up for the Strategic Machines slack channel
    b. Twitter – so the user can tweet a prearranged message about the platform
    c. Email – if the user would lie to subscribe for email notifications
    d. SMS – a web chat window which permits a user to subscribe to notification (this chat will occur with a 'default bot' – the microservice is not yet ready)

A constructor object has been developed to structure and simplify the process of interacting with the messaging platform, creating the json object for transmission to the platform, and interrogating the json object received. More information on the constructor can be found here

https://github.com/strategicmarket/message

This constructor is still being updated – but with every change semver is advanced so you can keep your project current with npm update. Docs are being improved as well

The constructor would need to be pulled in on the server side of webmachine – and routes developed for handling the processes and communicating with the messaging platform

Note that as part of the api, we will need to agree on how the jwt will be passed, and what other additional information about the user that you will capture and share (if authenticated, or if a guest). We can handle this requirement as a separate conversation when you are ready)

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 will be included as part of this deliverable
- Code will be delivered through pull requests on the 'strategicmarket' github site consistent with the codex.
- 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:

$480.00

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: