Meeting Minutes: SalesLoft (Private Sector)

Date: 07/27/2018

Time: 3:00pm to 3:30pm EDT

Attendance:

Butler Raines = VP of Products at SalesLoft (2.5 years there; prior API experience)

Bryan Hill = Software Development/Product Management

Joe Cosentino = Senior Leadership/Business Partner, Sprezzatura

Shane Johnson = Management Analyst, Sprezzatura

Nitin Sahai = Enterprise Cloud Architect, Sprezzatura

Thomas Holliday = Senior Manager/Microtask Lead, Sprezzatura

Interview Agenda Items/Questions:

General:

- Reached to you given you experience in this API space.
- What is your role in your organization?
  - Mr. Butler Raines' stated role is VP of Products at SalesLoft (2.5 years there; prior API experience).
  - Mr. Bryan Hill's stated role is Software Development/Product Management.
  - Website: https://salesloft.com/
- How many API's do you build each year? How many are for external consumption? How many for Internal consumption? What are the security concerns for internal versus external consumption? Firewall relaxation process and procedures.
  - What does your API infrastructure look like (I.e. how many you build/have built)?
    - Mr. Butler Raines: Private microservices working in the background powering the platform (how many APIs versus API gateways, etc).
    - Expose what, how many microservices (for APIs)?
      - Depends on where you are starting from. Might be starting from a monolithic system (I.e Web server and DB; costly to query that sort of thing). Breaking things off into a microservice is more cost efficient and efficient for the DB (access). If you have your API gateway set up, you can expose that externally. There is schema around versioning, using human (regular human speech) language.
      - Referenced Slide Share slide decks on this topic.
    - An end point is just a way to query a DB. Need to massage and transform the names before you give it to/hand over to a consumer.
    - What does YouTube's look like (example of a provider to look at).

  - Do you follow a top-down or a bottom-up approach for your API approach? What is preferred?
    - Mr. Butler Raines: Start with a customer request (I.e. pull ‘Subject’ line from a body of an email). API endpoints that customers need had to be created from scratch. Easier to expose things when you do not have to build a big framework on top of it.
    - They do not use Swagger (Mr. Raines is friends with the gentleman who started Swagger). SalesLoft created their own platform(s).
  - When you are approached by an App Team that wants to connect, do you have a specific UX person for the DEV folks to interface with?
    - Mr. Butler Raines: Have a product manager that is assigned to a key area of the platform. They work with the DEV folks to deliver a good experience. APIs are a holistic experience (endpoint design referenced; ensuring that your vocabularies are correct).
  - Whitelisting question:
    - Mr. Butler Raines: Look at SalesForce’s documentation. Referenced direct URL to keep your tokens fresh; OAuth referenced.
    - Mr. Bryan Hill: Trying to get more official, especially related to their public facing APIs.
      - Require an image. They then verify that what has been developed does what it says it is going to do. At that point, it is whitelisted.
    - Mr. Butler Raines: Developer account referenced.
    - Mr. Bryan Hill: Can build an internal application and not go through that (whitelisting) process.
  - Mr. Butler Raines: Have chosen OAuth codes and API key(s). It is cumbersome to build an OAuth2 application. Put API key in there because it is simple entry (get in quickly).
  - Industry standards followed/promoted/supported?
    - Mr. Butler Raines: Microservice is handled via their CTO through his (personal) contacts. Reference person (I.e. Swagger staff). Happy to introduce us to them.
  - Mr. Butler Raines: Likes Stripes APIs, Shopify. Mix Max tool in Gmail (likes that platform too).
  - API integration question and versioning (how do they handle that)?
    - Mr. Bryan Hill: Everything that they do has technical integration; on Version 2 now.
  - Any specific guidelines that you follow?
    - Mr. Butler Raines: If there is an error, you need to provide real-time error feedback.
    - Reference server logs.
  - Publicly facing API experience (I.e. hard lessons learned):
    - Mr. Butler Raines: Handling one off API requests without thinking out all of the different elements and angles proactively. Issue seen with over taxing the DB because of how things are coded (leading to the DB being queried and over taxed needlessly).
  - Thanks for your time. Important for the VA.

- Do you utilize external industry publications such as Gartner? N/A

Prioritizing API's to Build:

- How do you prioritize the API's you're building? Do you use an Agile Backlog, or other method? Do you evaluate based on Level of Effort and ROI, or use other metrics? Do you prioritize APIs as to their eventual utilization across the enterprise versus a single project?
    - Question not directly addressed in this section. See above.
- How much of your API backlog is defined by a Consumer's request? How much do you attempt to anticipate the needs of Consumers and build API's speculatively? Should speculative API development be more or less constrained by defined governess specifications?
    - Question not directly addressed in this section. See above.
- How do you make choices between building Experience layer API's vs Process layer vs System layer API's? Experience layer API's vs Process layer development could conceivably be performed by different development team – how will the governance model be enforced when there is a multi-team development environment?
    - Question not directly addressed in this section. See above.
- Do you allow Experience layer requirements to drive build prioritization on Process and System layer APIs? If not, why not? Or is the option of the building of "mocked" lower level API implementations (Process and System)?
    - Question not directly addressed in this section. See above.
- If you have API's/services on legacy platforms, how do you decide when port those over to a new platform? Additionally, how these legacy applications will be "ported" – re-hosted, re-factored, re-built?
    - Question not directly addressed in this section. See above.

Standards to Which to Build

- Do you have multiple, unrelated teams, delivering API's into the same environment? If so, what lessons have you learned about configuration management across the environment/teams? What level of maturity are your Agile, CI/CD, Dev/Ops capabilities?
    - Question not directly addressed in this section. See above.
- Does your organization maintain formal standards for: API Contracts, Naming Conventions; Version Control, Branching, and Merging; Testing Requirements; Exception Handling; Logging; Security? Would you be able to share any documentation? How is this documentation presented – web pages, WIKIs, Playbook SharePoint sites etc.?
    - Question not directly addressed in this section. See above.
- Do you experience a need to keep documentation and configuration control minimal/light? How do you ensure you're minimizing paperwork/bureaucracy and maximizing code delivery? Again Agile, CI/CD, Dev/Ops capability levels.
    - Question not directly addressed in this section. See above.
- How do you enforce standards across multiple development teams?
    - Question not directly addressed in this section. See above.

Consumer Experience

- How many customers external to your organization consume your APIs?
    - Question not directly addressed in this section. See above.
- What have you learned about development, testing, deployment that you apply to maximizing consumer experience?
    - Question not directly addressed in this section. See above.

Closing Questions

- Are there any internal documents (not published on your webpage) that you could share that would show how you apply governance internally?
    - Question not directly addressed in this section. See above.
- To whom do you look to for API governance best practices, with who else should we speak?
    - Question not directly addressed in this section. See above.
- May we please circle back with any follow-up questions?
    - Yes. Preference is for us to speak with Mr. Bryan Hill (more hands on experience with their APIs, technology).

Additional Reference Information and Analysis

Microtask Requirement

As the VA Lighthouse (now API Platform) Product Owner seeking the appropriate Governance model, I would like to understand, with the intention to adopt, best practices from the private and public sector, specifically for prioritizing APIs to build, standards to which to build APIs, and making the APIs usable by external consumers. We would like the primary research performed to gather best practices around:

- Characteristics of effective models in the public and private sector, and who is successfully using them.

  Including the utilization of leading edge API development enabling technologies (CI/CD, Dev/OPS, Micro service, Containers etc.).
- Lessons learned from these organizations (both what is working and what isn't)
- Highlight strengths and weaknesses of selected governance models.

____

Interview Analysis

Below are the observations from our brief call with SalesLoft.

Not forcing OAuth2 in Test Environment - SalesLoft discussed their practice of not using the OAuth2 protocol on API's in the public facing test environment. They did this to enhance the developer experience. Their believe was that it was burdensome for third party developers to build the flows necessary to implement OAuth2. This could be a prohibitive amount of effort for developers who simply want to test a SalesLoft API to determine whether or not it is worth committing the effort to develop their own third party application to interface.

> Analysis: We recommmend that VA consider this approach when developing its authorization governance model. The goal would be to enhance the developer experience in the public facing test environment. The trade off would be in the engineering resources required to ensure a version of an API exposed in test

does not required OAuth2 (or other) authentication. It would likely be prudent to publish a different version of the API test with authentication turned on in order to provide that validation prior to migration to production.

Verify Third Party Applications Function As Expected Prior to Whitelisting - SalesLoft reported that their Product Managers engaged in an effort to verify 3rd party applications prior to them being whitelisted. Once a developer builds their consuming application and tests it, they then contact the SalesLoft integration team in order to be whitelisted. The team will then perform some testing of the third party application to ensure it does in fact function properly with the SalesLoft API and data.

IRS has a somewhat analogous process. They publish a set of test cases for third parties to run in the IRS test environment prior to that third party being authorized to use production.

> Analysis: VA builds and exposes more APIs to be consumed by third party applications, it will likely become increasingly difficult to verify those third party apps function as expected for Veterans and other users. As part of its governance model, VA could consider how much governance it wants to place over 3rd party apps prior to them being whitelisted in production. For example, is there a requirement for VA to itself test some third party applications? We anticipate a trade-off here between making it easy for developers to deploy their own apps consuming VA data and imposing a heavy testing and whitelisting process. We recommend VA builds this aspect of governance incrementally, with the input from any Product Managers or Customer Experience Engineers it retains.