VA Lighthouse Governance Model Microconsulting

Ben Kutil

Ad Hoc, healthcare.gov API team

6/22/2018

## Warm up questions

* Tell me about your role and responsibilities.

Acting as product lead for Healthcare.gov.

* How do APIs fit in?

In relation to APIs, Ad Hoc delivering and maintaining 4 diff. APIs using healthcare.gov. APIs are made availa. To 3rd parties typically healthcare issuers, or direct enrollment companies (helps with enrolling people into healthcare plans). We maintain the authentication process and ability to give an issuer or direct enrollment partner an API key or the ability to rotate that API key, so they have access to pull that info out of the marketplace

* Deciding what APIs to develop: center for medicare and Medicaid services and the group within that has an emphasis on building all their tools API first and with an emphasis on making data public as much as possible, so that private companies can build on top of it. Not only to offer their government services but to create an additional competition for those services in the private market. So when working with data the org has outlined that they want to make public, that’s one way we start to say let's make an API to do that. Multiple tools that make use of the same data another opportunity or signifier that says lets build this into its own service so multiple tools can make use of it. Ex: Marketplace API- database of all the health plans used in healthcare.gov, database of all the providers, drugs and facilities those health plans cover and business logic around all of those costed out or calculated and we use all that info within healthcare.gov across 4 or 5 diff applications. So instead of all those applications duplicating that info, we built an API to allow each of those tools to make use of it
* Consumer needs: it’s essentially transparent whether we had this data by API or a tightly integrated application stack, .a consumer would not know the diff. , the larger end-user audience has sort of been defined for the marketplace so the issuers who are involved and the 3rd party that might be developed was really building it as an API or having the APIs available creates those opportunities for them to build tools more easily
  + If not familiar explain APIs: An API, or Application Programming Interface, is a flexible way to give and receive information by allowing software systems to communicate with each other. In this case, the VA could provide a set of tools for providing appeals data that gives your developers all the building blocks necessary to automate collection of appeals data directly from VA’s system to your own.

## Best Practices and Governance Models

* *Developer:* How do you decide what APIs to develop? Can you walk me through that process?
  + How does that relate to end-user needs?
  + How do you assess whether an API is ready for production?
  + What’s been your experience with API standards or approval processes?
  + How have you rolled out standards with APIs (esp in a bureaucratic setting)? What was that like?
    - Best practices: versioning APIs
  + How do you balance standardization across an organization while allowing for flexibility and innovation?
  + Where it has gone wrong: CMS is in the process of helping 3rd party like health..or others take over or have more of a say over the application process for healthcare as well. An API for submitting apps, that API does follow some standard in terms of how paymode info is organized and the types of values that its expecting but it doesn’t follow standard around versioning APIs, which is considered a best practice and that has severely hampered the development team working on our internal marketplace app as well as the 3rd party partners who are trying to get around their own app process on top of this API. It hampered them b/c everytime this API new changes that break existing functionality or changed existing functionality it breaks everyone's apps, so its slowed down the flexibility or ability to innovate in that everyone has to have lock-step deployment process to account for the changes they are making. Typically an API would follow a versioning best practice , where there will be a version endpoint that was stable so all the groups using that API could dev..against it and have their flexibility and the team whose developing that API can develop the next version independently of all the teams using that API.
  + Ex: marketplace, CMS is helping 3rd party...take over the app. Process for submitting applications…..– standard alledth open API, itself generates documentation, where you would send a request for info, as a developer team that manages the API or end-point…. As its being developed the best practice as for documentation, ex: call the JCON ..schema- respond to requests, versioning deprivation and other examples of best practices
  + JCON API schema- provides standard way that people can structured their responses and requests. The benefit is that it reduces dev complexity in that you always know how your request is going to come back.

· Versioning and versioning deprecation scheduler are other best practices

* *Developer:* What’s been your experience with implementing APIs and what problems have people had with it?
* Adopting new standards, it hasn’t been challenging, that part has been relatively easy. Versioning best practice has been a challenge because it's not a group we work with and it's not a group that our partners at CMS have sway or control over.
* **Challenges**: the smallest one is developer frustration and the impact of that frustration on the overall development timeline, if an API is not following a consistent structure handling the payloads going out or coming back is frustrating and can impact development time.Same for documentation can impact development time and adoption (ex: learning about a new API but there is no documentation, incomplete or cannot easily try out request and see if I can get responses, this limits interest someone may have in using or adopting that API versioning and API for applying that is going to fundamentally affect the dev timeline of all the apps and tools using that API. They will have to coordinate schedules to account for breaking changes. Those are the two biggest things that could impact not only individuals but entire orgs. As they try to develop against that API

**Best practices**: standard called the open API format, represents a documentation standard, benefits of this is that it self-generates documentation as you're developing an endpoint. This is where you would send request to get info as the dev team maintaining that API instead of having to write up documentation on how to use that particular functionality or endpoint. Its generated out of the code as its being dev. Many tools can use that API format to generate testing tools (ex: swagger, can ingest the open API format and turn into documation as well as self running testing tool, where a person can test endpoints of the API

## Authentication frameworks

* For any APIs you build, how do you manage security/privacy concerns?
* What are some things to keep in mind when creating/working with an API that manages authentication?
* The one benefit is that APIs are read only. People are submitting their personal info into the system, which simplifies the security requirements. Data is considered public (also simplifies application development security requirements), have an API key and ability to set API keys for certain limiting access but that’s not so much for security because most of our info is public so it’s more for analytics purposes and ability to maintain traffic and volume
* How familiar are you with identity proofing solutions?
* The authorization and authentication are custom built solution, we are looking to moving it to something like Octa or CMS as a larger identity tool we can interact with. Why?? Because right now the Healthcare.gov authorization system manages its own database of users, pull from multiple databases of users and its long term goal to unify to healthcare.gov and medicare.gov account process.
* A consumer would just have two accounts (just healthcare.gov and myMedicaid.gov) Buying an Identity proofing system: the marketplace already has integrations with DOH, IRS and Social Security Admin they have built their own as part of the original healthcare.gov development that was the identification process that was built, would probably continue to use that. Prior to starting an app there is remote identity proofing step that calls on those 3 services. They control the look and feel of that. It’s a format of “you have previous lived on what 3 streets pick the select it”
* They do not have multi-factor, its low priority for them due to people only logging into their service once a year and they already have a hard time remembering their password, that causes another layer of complexity and frustration to redoing their healthcare. Would rather have a link they can identify with
* The authorization and authentication are custom built solution, we are looking to moving it to something like Octa or CMS as a larger identity tool we can interact with. Why?? Because right now the Healthcare.gov authorization system manages its own database of users, pull from multiple databases of users and its long term goal to unify to healthcare.gov and medicare.gov account process.

A consumer would just have two accounts (just healthcare.gov and myMedicaid.gov)

We authenticate the call center reps who work on behalf of consumers

## Closing Questions

* Is there anyone else that you think would be good for us to talk to?
* Is there anything we haven’t talked about today that you think I should know?
* Any last thoughts?

Additional Qs:

3rd party question- We do authenticate the call center reps who work on the behalf of the consumers and their info comes out of the database the same way whether it’s a call center rep or a consumer

How can you see the applicant’s data? From the applications point of view the API request are always the same, the authorization and the authentication happens within the application itself to request data (No framework around it if someone has a power of attorney)

All consumer accounts are a 1:1 relationship And all other support accounts are a 1:many relationship

The call center and a few other advanced revolution roles all have access to any consumer records that they are working on

Applying on behalf of a Parent: person who applies is considered the primary contact, its considered their account and in the application process they will identify themselves as a non-filer (then the application is not for them but who they are filing for)

How can you make sure that person is allowed submit an app on someone’s behalf? That is part of the account creation process (LOA2)

Who else should we talk to?

Corbalt - used to be Super Brilliant - they’ve been working on a lot of the api management tooling. James@superbrilliant.io

NAVA- authorization and user accounts