

## **Meeting Minutes: Forest Service (Public Sector)**

**Date:** 07/27/2018

**Time:** 10:30am to 11:30am EDT

### **Attendance:**

**Albert Ahr** = Application DEV/Branch Chief for pp DEV group (in CIO area of FS); Enterprise Business Solutions Services (DEV by trade)

**Shadat Mahmud** = Solution Architect/CIO Office (Enterprise Application DEV branch)

**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

**Nolan Ventre** = Project Manager, Definitive Logic

### **Interview Agenda Items/Questions:**

#### **General:**

- Introductions conducted to start the call.
- Mr. Joe Cosentino: VA has a complex internal IT environment. VA is trying to mature its experience with exposing 3rd party data (etc) external to their organization. Interested in API governance, implementation patterns, 3rd party DEV experience.
- What is your role in your organization?
  - Mr. Albert Ahr stated role is an Application DEV/Branch Chief for DEV group (in CIO area of FS); Enterprise Business Solutions Services (DEV by trade).
  - Mr. Shadat Mahmud stated role is a Solution Architect/CIO Office (Enterprise Application DEV branch).
  - Website: <https://www.fs.fed.us/>
- Forest Service API environment today:
  - Mr. Albert Ahr: Been with the Forest Service for 2 years in the fall.
  - In terms of (API) DEV, all based on services and API DEV. Exposure outside of the Forest Service. Use cases for that?
    - Mr. Shadat Mahmud = As Mr. Albert Ahr mentioned, we do have use cases for microservices and APIs (e.g. system integration side of the house). Primarily internal. Two main external uses cases = Geospatial and collaboration with Department of Interior. Trusted communications with HR systems. Public facing = Only one is Geo 'special' (they do fire incident and claim management).
- 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.
  - How many with prioritization?
    - Mr. Albert Ahr: It is substantial. Prioritize on a system-by-system basis.

- Based upon client requirement? User experience? API internally exposed (I.e. lookups, system to system calls, etc)? Capability APIs question too?
  - Mr. Shadat Mahmud = Geospatial branch: Map services out there utilize Forest Service APIs for their mapping.
  - Additional data points shared:
    - 3K APIs utilized with/for Forest Service.
    - Trusted communications (as needed).
  - Over the last year or two, the mantra is to be user experience driven.
  - Need to go through APIs, and use GitHub to share data and services (including use of GitHub for internal communications). Minimized DB POAMs by following this model. Able to modernize internal stack because they are not beholden to a particular DB.
  - Have been looking at some API governance tools. Bought one of the products last year (after engaging in various pilots).
    - Included lifecycle and version management, SLAs management.
    - Took a step back this year on purpose (I.e. did not renew the product license this year). The vendor's tools required purchasing other tools (increasing expense).
    - Have tested with multiple vendors, and are now looking for another vendor. Automation and containerization mentioned; some of the maturity is not quite there yet.
  - More specifics on the tool? VA has gone down the IBM route (which is very expensive). VA now looking at a MuleSoft solution and other solutions.
    - Mr. Shadat Mahmud: Lots to share! Add us to their market research.
    - IBM has matured quite a bit, but it does take a lot to run these types of products. Too much time would be spent on integrating the products.
    - ASB with SW AG (missed the meaning of these acronyms)?
    - Felt that business view was not there; too technical focused.
    - Should be able to utilize the tools without a focus on too many technical details. Customization on the business side was missing.
    - Brought in CA tool. Felt it was very solid. Another group using the CA tool too. Enhancing their product a lot as of late; hard to keep up.
    - Document container (implement). Six different products to get that to work.
    - Forest Service not planning to that level, so passed on that one (IBM?).
    - Looked at MuleSoft, but did not feel that they were mature enough yet.
    - Looking at the Google cloud too.
    - MS Azure is what they are looking at on the cloud side.
      - Mr. Albert Ahr = Everything that has been built is cloud native methodologies. Once the governance is in place, move to the cloud (I.e. MS Azure focus; yes).
  - Everyone should be able to DEV APIs. Need to follow a set of standards (limit 'garbage in and garbage out').
  - Mention of 'cloud native side of the house.'
    - Cloud native type solutions are a focus right now.
  - Various programs going. Own runtime? Various DEV Teams approaching this as service owners. CI/CD pipeline and one Runtime under the hood? Or, something different?

- Mr. Albert Ahr = Forest Service is like the VA. Huge in terms of their portfolio. They can speak to the CIO portfolio (can speak for that piece).
- CI/CD pipeline is mostly technology dependant as to the stack that the DEV teams are using.
  - Mr. Shadat Mahmud= Not API specific. Pulling data from GitHub. Automation is a focus.
  - Standard CI/CD pipeline = Use GitHub and Jenkins (as the platform for CI/CD along with GitHub). Technology independent; have done work in this area.
- Mr. Shadat Mahmud: Change management, security (etc) are all important to API governance for Forest Service.
  - There are SLAs that they have and envision to help maintain security, etc.
  - Not taking any more SOAP services. Work with Cyber on security policies.
  - Want the API community to experience an ease of working with Forest Service, but there are rules to follow.
  - 'Language agnostic' mentioned.
- Use of microservices: What goes in the API layer?
  - Mr. Shadat Mahmud = Easier said than done. Want users to use Agile practices, etc.
  - Very reactive at times. 'Learn and grow' model is being followed today.
  - GitHub = UI/UX, APIs, Data Services (various 'projects' that they are running concurrently right now). APIs are an independent element for each of these projects.
- Lessons Learned: API 1st approach (legacy, as well as new).
  - Mr. Shadat Mahmud = Legacy ones; Very legacy (they do not waste a lot of time to address those). Provided an Oracle example (cost, license impact, etc). Needs to be demand based, use case based and make sense. Less than 100; some sort of criteria (I.e. SQL server, etc). Been practicing quite a bit. API based DEV is easier.
    - Legacy APIs/Systems/Apps = Unless there are new features that are coming in, then it is left alone or forcing folks to move to APIs (versus Oracle forms). DEV is being pushed to APIs because the legacy approach is not sustainable. Stopping at the firewall level even.
    - Mr. Albert Ahr = API DEV is prioritized via 'targets of opportunity' within their portfolio (I.e. outdated tech creating security vulnerabilities). Rare that Forest Service would take something that is old and try and architect new services around it.
  - Do you version the legacy elements? ALL or none approach, or gradual change?
    - Mr. Albert Ahr = Almost always a complete switch. A lot of their elements are seasonal based (referencing the mission of the Forest Service). Can do switchovers during the off season times (I.e. hard cutovers in the off seasons).
- Any playbook or documentation for their APIs?
  - Mr. Shadat Mahmud = Documentation should be next to the API in a governance tool. Forest Service using GitHub for this right now.

- API governance tool would be most beneficial to end users and API DEV folks. During runtime, could come back to run a report to see how things are utilized, etc. (I.e. see which ones are being used more than others; limiting impact as to what they are focusing their time and attention on).
    - GitHub primary engine for 'intersourcing'; maybe utilize a Wiki?
    - Source code policy mentioned. 'X' amount of gov't source code data should be shareable with US citizenry.
  - How do you govern APIs once the data has been shared with the public?
    - Varies by organizations.
    - Not 100% there. Know where they need to be, though.
  - O&M perspective = Any quality of service issues (across the enterprise, etc)?
    - Mr. Shadat Mahmud = In theory, should not have any issues during runtime.
    - They do run into issues. Majority of system is not elastic (like the cloud).
    - Trying to move to the commercial cloud. Doing a lot of pilots to get ready for this. Once on the cloud, should be able to tackle the runtime issues/concerns.
    - Usage related feedback = How many people are using a particular App?
      - Servers went down due to constant calling out to the API. In the header of the API helps with the audit trail. Found out who was doing the rapid, multiple callouts (I.e. loop calling with performance testing; Forest Service does not have the resources to deal with that sort of thing).
- Do you utilize external industry publications such as Gartner?
  - Worked with Gartner and others.

#### **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 addressed specifically in this section of the meeting notes.
- 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 governance specifications?
  - Question not addressed specifically in this section of the meeting notes.
- 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 addressed specifically in this section of the meeting notes.
- 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 addressed specifically in this section of the meeting notes.
- 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 addressed specifically in this section of the meeting notes.

### **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 addressed specifically in this section of the meeting notes.
- 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 addressed specifically in this section of the meeting notes.
- 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 addressed specifically in this section of the meeting notes.
- How do you enforce standards across multiple development teams?
  - Question not addressed specifically in this section of the meeting notes.

### **Consumer Experience**

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

### **Closing Questions**

- Are there any internal documents (not published on your webpage) that you could share that would show how you apply governance internally?
  - Sharing of playbook? API Governance perspective.
    - Shadat = Have produced an API governance document (not really a 'playbook'); will send to us today.
- To whom do you look to for API governance best practices, with who else should we speak?
  - Question not addressed specifically in this section of the meeting notes.
- May we please circle back with any follow-up questions?
  - Mr. Albert Ahr/Mr. Shadat Mahmud = OK with follow-up. Yes.

## 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 noted observations and analysis from our call with the US Forest Service.

**Ancillary Tool Requirements to an API Gateway Tool Can Add to Licensing Costs** - Without getting into specific tool pilots, the Forest Service noted that they witnessed licensing costs increased by ancillary tools required for API Management. They noted that many API Management tools on the market do not offer all of the offer full functionality. There could be different tools for the Gateway, queues for microservices, monitoring, SLA enforcement, etc. The expectation from vendors being that users by other tools from the offering's family to round out functionality. Forest Service noted that they used Gartner research and their own pilots to examine different tools.

Analysis: VA currently uses a variety of tools for API and microservice platforms, gateways, etc. and has conducted its owns studies of what to acquire over the previous few years. It is not clear that there is an obvious answer for large federal agencies based on the VA and FS' experience. Regardless, a single API directory and gateway would appear to be a requirement for ease of use for 3rd party developers. Early governance activities could focus on making a decision for an enterprise gateway tool, factoring in lifecycle cost considerations. We anticipate this being a challenging decision requiring focused resources to properly account for trade-offs.

**Purposeful Division of Teams on Github to Enforce Architectural Goals** - Forest Service reported that they used Github as a governance structure for enabling their target API and Microservice architecture. For any project, they forced the division of the project into three teams on Github: ) UI/UX; 2) API Development; and 3) Data Services/MicroServices. This helped ensure a microservice architecture on new initiatives.

Analysis: This is interesting from a VA perspective given the amount of effort put into building the current enterprise schema in Rational as well as the mandated use of Compliance Epics. To our knowledge, there

is no incentivizing of an API or microservice approach within the VIP requirements. VA could consider amending VIP and the Rational schema/Compliance epics to encourage API's and microservices. This could potentially be piloted within Enterprise Shared Services or encouraged on any new project using Github.

**Reliance on Non-Cloud Based Legacy Systems Decreases API Availability** -The Forest Service confirmed that they do have experience API uptime issues associated with API's built on top of legacy data stores not yet located to the cloud. They observed that cloud based systems are more reliable resulting in better uptime for associated API's.

Analysis: We see the governance implications for API development in the VA domain as two-fold. First, as VA initiates its own Cloud migration initiative, it may be worth factoring in the API pipeline when considering prioritization of legacy system migration. That is, upon examination of the API build pipeline.