LESSONS LEARNED FROM LIVING IN AND AROUND VC-BACKED SOFTWARE STARTUPS DELIVERING DUAL USE DEFENSE TECHNOLOGIES

REFLECTIONS BY CECE SMITH

One of my concerns during my evaluation of new defense tech startups has been assessing the company's realistic *readiness for* and *commitment to* the federal defense market. I have fallen in love with the world of defense tech software startups. It is a niche tribal community of innovators working to rapidly bring best-in-class commercial-off-the-shelf technology to a multifaceted world of mission-centric end users often blocked by the multi-layered minutia inherent in bureaucracy.

However, as I have observed from my past experience as well as through watching the market, venture capital / VC-backed startups serving up commercial software with dual use defense applications have unique challenges that differ from traditional private defense-only or defense-first (then later commercial) contractors.

- We have a commercial-driven product roadmap.
- We have different challenges and boundaries to confront when we need to adapt that technology.
- We have a set of global team members with and without clearances and from countries all over the world.
- We have different Intellectual Property and Technical Data Rights protections to assert based on technology we previously developed either exclusively or primarily at private expense.
- We have different acquisition pathways available to us; as well as some that are also *not*, for a vast variety of reasons.
- We have different regulatory compliance requirements designed to protect defense technologies, especially against influence from foreign investment and foreign government instrumentalities, although many of our advanced technologies have been developed because of our global reach and international presence.

The list goes on and on…

So successfully surviving, deploying and scaling software products across the Department of Defense (DoD) as a startup is notoriously hard. The technical complexity and nuance of deploying software for defense requires rigorous and sustained surgical precision. Whether cloud-hosted compute or locally-managed, bare metal infrastructure, collectively these networks are designed to intentionally target,

prevent, intercept and expel commercial software as though they are foreign objects invading the body.

But it is exactly *this* that makes the challenge both equally competitive and rewarding. To work with defense customers, it is critical to meet the end users where they are, which really demands deploying software to an environment where it can be accessed at an end user's (most often) classified workstation. For commercial product-driven startups whose software architecture was originally crafted on a public / commercial-cloud hosted environment, such a deployment endeavor requires developing a stand-alone version of the software that can meet the rigorous requirements necessary to initially survive and then maintain support for end users operating in a classified environment.

Founded by Amazon Web Services (AWS), the infrastructure on the Commercial Cloud Services (C2S) contract vehicle has supported the unique cloud computing needs of the U.S. Intelligence Community and their modernization roadmap for the past 15 years. The first cloud services provider (CSP) to offer compute infrastructure on all regions of the U.S. Government's varied classification requirements (serving federal, state, and local needs), by sheer timely speed, AWS has arguably built a powerful moat around their government portfolio; including integrating themselves into the federal compliance framework by facilitating Cloud Services Offerings (CSOs) trying to obtain Authority To Operate (ATO) approvals like FedRAMP. Some argue that the 2020-awarded Commercial Cloud Enterprise (C2E) contract threatens to slowly unseat AWS' position, by introducing a "cloud agnostic"[1] acquisition vehicle supporting more CSP optionality – including Microsoft Azure (which has held influence in the armed services), Google Cloud Platform, IBM and Oracle.

Over the past few weeks, my conversations with many of the Applied Intuition team has centered upon my past experience, much of which has been more focused on machine learning around geospatial datasets, the platforms supplying that imagery, and the software for serving up these datasets of value to Defense and Intelligence end users. These enjoyable conversations have catalyzed my own personal reflections[2] on lessons I've learned from living in and around software startups accelerating innovation for dual-use defense applications.

[1] I have a lot of opinions on what this means (if anything at all) in the context of government networks.
[2] All of the following reflections, information and anecdotes are unclassified.

Case Study: Difficult Yet Valuable Lessons Learned from Simulating and Successfully Deploying Software on AWS C2S

My favorite mission-focused programs were collaborations with computer vision, data scientists and software (Middleware and DevOps) engineers executing robust DoD requirements:

- Developing computer vision algorithms that detect and monitor rare objects of interest. Multi-fusion computer vision models were trained to detect and classify objects within unclassified commercial data and then classified imagery and full motion video (FMV) sources across multiple sensor modalities (e.g. detect and classify objects such as types of aircraft, on multiple modalities, including High Altitude Long Endurance (HALE), Intelligence, Surveillance and Reconnaissance (ISR) Unmanned Aerial System (UAS). Because some of the objects of interest for defense customers are rarely observed, strategies evolved to include generating and incorporating synthetic data to increase the technical feasibility of meeting mission objectives within condensed timelines.

- Adapting and delivering standalone versions of commercial AWS-hosted platforms for deployment to secure / classified environments. Out of the aforementioned quests, this was *by far* the most challenging and technically-rigorous pursuit.

My responsibilities included delivering comprehensive Design Documentation based on DoD requirements, followed eventually by a Test Plan describing the test environment, test procedures, quantitative comparison tests, and the test data and materials used by our development team to verify that the software solution not only met the DoD's requirements, but was also fully functional. The DoD (for these programs, oftentimes cadres of multiple Defense and Intelligence Community Customers participating in a set of programs or with participating groups of end users within a joint-agency collaborative construct) required multiple Test Report deliverables explaining the selected testing methodology in sufficient detail for end users of varying technical proficiency levels (from end user to Certified Information Systems Security Officer / CISSO) and mission user personas (geospatial analyst, imagery analyst, warfighter, machine learning specialist, etc..) to not only reproduce but also validate the test results using the data and materials provided with the Test Plan. Tasks included: software installation; testing specific mission use cases on the software solution; administering and provisioning user access, roles and permissions; testing automated data ingestion pipelines; testing push and pull tip-and-cue / alerting workflows via a developed RESTful Application Programming Interface (REST API); deploying containerized computer vision and data science normalization algorithms to a hosted

services environment; running / deploying algorithms on ingested data, and testing and evaluating algorithmic performance[3] based on output results. These programs culminated in a live demonstration of the end-to-end installation, configuration, and application of the software on pre-identified use cases on randomized test or real live data.

These multi-phase, multi-year programs were truly a gift; they arguably taught me the most I've learned to date about delivering software to meet Defense and Intelligence Community requirements. In particular as relevant to some of the discussions I had this week, iteratively writing Test Plans required thinking about and describing new functionality in a format that could be easily translated into verifiable test requirements on geospatial datasets.

Learn to Love and Embrace Living in Uncertainty

There are several "unknown unknowns" inherent in conducting C2S software deployments. Deploying a commercial cloud-hosted software platform – particularly any solution designed to ingest data from one or multiple sources and modalities for computer vision processing, requires a fundamental change in system architecture. While a solution might be a Software as a Service (SaaS) offering for commercial customers leveraging unclassified / public AWS, for defense end users, that same solution may suddenly become a Platform as a Service (PaaS) solution consisting of several custom applications interacting with other ones. Arguably, it may be more of a pseudo-PaaS solution due to the nature of how cloud-hosted compute infrastructure is provisioned and managed through a collaboration relationship between Cloud Service Providers and government. Navigating the consequences of this reality taught me how to embrace true uncertainty and that the magic of solution design really lies within its contextual *constraints*.

To that end, succeeding as a nontraditional defense contractor requires not only expecting, but *running* head-first into a world of programmatic and technical dependencies that constantly compound the number and complexity of unknowns and uncertainties. We depend on resources, account access, information, tools and support provided and often provisioned by multiple government departments, agencies, and even other prime defense contractors. We not only need to have a mitigation strategy lined up in advance, we also need to maintain our "risk register" that anticipates, forecasts, rehearses and simulates the likelihood and magnitude of the impact for identified programmatic, technical and business risks.

[3] More on this later...

Software Deployment & System Integration Risks

Although their public dialogue always inspires much dialogue about joint-collaboration, each Defense and Intelligence Community Customer has their own custom user parameters, data pipelines, data sources and platforms, acquisition vehicles and programs, compute task orders administration processes and software integration requirements relevant for their own secure environments.

Technical requirements for most of my machine learning programs included developing ingestion pipelines capable of delivering Government-provisioned data sources of varying classification levels and modalities (as well as the essential training data for machine learning algorithms) into a platform solution. Because these data sources are provisioned by different agencies and commercial / defense contractor vendors, they are all functionally "external sources" providing inputs – and in some cases – obtaining outputs – to and from the platform. Even *just initiating* conversations with anyone responsible for provisioning these pipelines required developing digital engineering documents based on the system markup language (SysML) Department of Defense Architecture Framework (DoDAF) views, including:

- OV-1 High Level Operational Concept Graphic
- OV-5B Operational Activity Model
- SV-1 Systems Interface Description
- SV-4a Systems/Services Functionality Description
- OV-1 For AWS Deployment

Although all of these captured diagrams started out as notional artifacts, they evolved and helped shape conversations and collaborations with other government and contractor technical professionals prior to and throughout the program. Operating software in the defense environment demands inherently systemic thinking; any solution always represents one individual or set of stops within a greater (behemoth) pipeline, in a robust environment supported by a diverse ecosystem of government and commercial tools, sensors and hardware and software platforms. The necessity of interoperability or "playing well with others" in these environments persistently impacts everything from APIs, control flow, communication protocols and connections, standards conformance (of which there may be many, on any given use case), data flow, data formats, data storage techniques (and available options), schema, infrastructure and compute requirements, and impacts to scalability.[4]

[4] As well as Total Cost of Ownership (TCO) for that defense customer…This was a common compute battle.

Standalone versions of a software platform are not permitted to have any dependencies or connectivity outside of the C2S Region (e.g., access to any open internet addresses). In cases where the software is a dual use adaptation of a pre-existing commercial cloud-hosted software product already serving commercial end users, meeting this requirement demands creative strategies to address – and essentially replace – dependencies. Little things I had taken for granted – like leveraging third-party Software Developer Kits (SDKs) to power the maps and location data that served as critical components both collecting user inputs from the graphic user interface (GUI) as well as visualizing computer vision outputs over an area of interest for geospatial use cases. Replacing commercial location data with government-provided or government-provisioned data was an easy substitute; making sure that:

- global mapping for any area of the world was available and represented in the GUI
- those maps were accurate, reliable and conformant to geospatial data standards
- those maps could feasibly be regularly updated; integrated into maintenance release cycles; and documented in release notes that required prior approval and notice of any third-party vendor packages
- the maps could be provided to customers in advance for software installation and configuration on an appropriate media format that similarly required no internet access or downloaded content from the internet

Scaling a program offering access to unclassified software capabilities to production-level integration within secure environments accessed at multiple user workstations, in my own words, feels like playing a multi-layered video game laden with many villainous traps. We might 'die' a few times, but it doesn't matter as long as we persist and keep playing. It is the iterative resilience of playing over and over that teaches us how to out-trick the traps. Preparing requires (again, iteratively) conducting a deployment of the stand-alone prototype to one or more third-party AWS C2S simulated environments designed to help innovative companies anticipate differences and remedy adaptations required to operate software in an air-gapped network. For example, my programs each dealt with the differing availability of AWS components across regions supporting workloads for varying classification levels as well as ensuring that any of the components available on all regions were shored up to require no connectivity to the open internet.

Software Stack Security and ATO Landscape: The Only Way Out is *Through*.

As nontraditional defense contractors, we are charged with continuously "herding the cats' and orchestrating a robust network of DoD security Points of Contact (POCs) handling the required documentation and workflows necessary for obtaining government approvals.

As part of the operational security posture for their information systems, many government systems that defense-focused software companies will encounter require that the personnel performing the 'hands-on-keyboard' work attain and maintain specific network security certifications or credentials. These may vary across systems and include accreditations such as becoming a Certified Information Systems Security Professional (CISSP). Engineers on some of my programs had to obtain Information Assurance (IA) certifications to a certain proficiency level (e.g. Information Assurance Technical - IAT; Information Assurance Management – IAM).

Strategically scaling growth and accelerating software adoption across the defense landscape also requires navigating acquisition-focused compliance requirements found in Defense Federal Acquisition Regulation Supplement (DFARS) as well as technical requirements with acquisition impacts, such as National Institute of Standards and Technology (NIST) compliance and the recently released Cyber Security Maturity Modeling Certification (CMMC), actively serving as a "guard at the gate" to prevent past after-acquisition afterthought considerations of security requirements.

These levels are annotated in the Joint Personnel Adjudication System (JPAS) for managing account accesses and clearances – and will be visible / verifiable to anyone with JPAS or Defense Information System for Security (DISS) access. Due to the latency introduced by training and testing requirements for certifications as well as the lag time for certs showing up in JPAS, asking about documentation and personnel requirements early and following up often is a critical component of managing software programs serving defense use cases.

Hosting Technical Exchange Meetings (TEMs) offers a strategic opportunity to discuss the engineering and provisioning processes required to gain access to government networks, accounts and workstations. Unfortunately, dual-use capability startups that dip their toe in the defense world often encounter challenges obtaining or maintaining actively-held clearances – sometimes reliant upon subcontracting arrangements or temporary / inherently unscalable pathways through rapid acquisition and defense innovation programs. But scaling sufficient impact to bridge the "value of death" demands the ability to obtain a Facility Clearance (FCL) allowing the company a bit of autonomy towards holding active and preventing infamous 2-year lapses in clearances for their own personnel. I've learned and observed the value of advocating for DD-254

sponsorship (a prerequisite government form for justifying an FCL based on the program's requirements) early on for any requirements that exhibit high potential for veering into the classified space. Sometimes kicking these conversations off early also facilitates an opportunity to leverage joint-agency programs or use case synergies to expedite sponsorship by latching onto in-progress programs.

Navigating Nebulous Compute Nuances

Understanding how compute infrastructure is provisioned and managed in the DoD can make a meaningful impact on the momentum for executing technical requirements. As mentioned earlier, at the time of my program, Commercial Cloud Services (C2S) served Amazon Web Services (AWS) into the Intelligence Community (IC) through a single contractual agreement and as part of the larger IC Information Technology Enterprise (IC ITE) joint cloud environment. My role required that I work with Intelligence Community POC for cloud services, responsible for working with internal government customers to coordinate AWS C2S services through agency or unit local contracting offices. It is through these engagements that I quickly discovered that compute and infrastructure management for systems of systems frameworks is incredibly nuanced and complex.

On some of my programs, there were often multiple avenues to obtain compute, as well as multiple available (whether happily willing or not[5]...) sponsors. But it was even more critical to find the *most appropriate* sponsor based on our technology and the end users it served. I also encountered stakeholders with varying understanding, opinion and willingness to advocate for the most optimal compute environment supporting end user requirements. For example, one stakeholder strongly advocated that we conduct a bare-metal, on-prem deployment of the software to a locally-managed workstation physically connected to secure networks. Doing so would expedite our ability to prove out the mission use cases, give access to end users, and bypass a cumbersome and lengthy Authority to Operate (ATO) process. However, after collaborating across our engineering team, defense end users and IT professionals from multiple agencies, I realized I needed to rapidly compel the stakeholder to reconsider. Although the on-prem idea initially sounded less risky, it threatened our ability to deliver the strategic outcomes of the program – namely by failing to reliably provide sufficient compute required for processing petabytes of data; hosting the software, computer vision algorithms and training data; reliance on a manual versus automated ingestion pipeline of data provisioned by multiple other agencies; and lack of scalable accessibility – the critical user adoption necessary to make the product "sticky" to strategically grow the account. This anecdotal example demonstrates how critical it is

[5] Perhaps "accessible" is more appropriate than available...

to identify the most appropriate – as well as strategically advantageous – avenue for gaining access to compute infrastructure. By identifying an existing and available or executing a new C2S Task Order early on, startups can ensure they minimize risk and stakeholder concerns by obtaining a sufficient amount of compute infrastructure for hosting, developing, integration and disseminating / provisioning access to software technologies.

The Ride Never Ends...Managing Releases & Maintenance

There's the hurdles of initially deploying software to defense customers, and then there's *maintaining* that software... For non-traditional defense contractors, providing updates, minor and major releases and release notes, and bug fixes minor requires the additional complexity of ensuring that:

- Updates released to the commercial cloud-hosted software solution consider the Defense version of the solution, its dependencies (many of which may differ), and its intended release schedule.
- Testing of aforementioned releases and bug fixes in a simulated environment prior to deploying those updates within the customer's environment.
- Anticipating that those releases will likely encounter bugs and require fixes / updates once deployed to the customer's environment – even if those updates passed the C2S or other simulated environments.
- Preparing to deal with dependencies on other Government and commercial programmatic and technical actors that play any role in overseeing, administering, provisioning, integrating, securing and / or maintaining the software.

Maintaining Momentum By Orchestrating a Complex Cadre of Capability Collaborators

Some of my programs[6] centered upon researching, innovating and then productionizing AI-enabled systems to detect and monitor illicit activity, anomalous patterns of life (PoL) and infrastructure security threats. To help this set of customers strategically manage human-deployed resources, the system leveraged multiple autonomous and space-based platforms in a series of tip-and-cue frameworks designed to prioritize additional assets and human intervention and / or actionable decision-making based on computer vision detections on data derived from the aforementioned platforms.

[6] I learned a lot from working on this same set of programs at two different companies and vantage points but in similar roles

Developing the system relied upon multiple components provided by collaborations among multiple companies:

- 10cm imagery from stratospheric balloon robots to provide high-resolution, high-frequency imagery over areas of interest (VC-backed commercial startup)
- A deployable High-Altitude Balloon (HAB) platform, selected for its rapid maneuverability, long-duration mission / minimal resupply capability, and its ultra-persistent wide area communications supporting near real time delivery (traditional defense contractor).
- Tethered Aerostat Radar System (TARS) as a low-cost yet wide-range surveillance system to detect suspicious or illicit aircraft flying without a transponder signals (traditional defense contractor).
- Radio Frequency (RF) data derived from space-based small satellites to detect geolocation signals from UHF Push-to-talk radios and L-Band mobile satellite communication devices (VC-backed commercial startup).
- Geolocation data from non-aerial commercial sensors (VC-backed commercial startups).
- Sensors and Platform Components borrowed from another classified Government program and mounted on the High Altitude Balloon platform to deliver data over areas of interest at a resolution deemed likely the most technically feasible for detecting features of interest (UARC).
- AWS Ground Station capabilities borrowed from another classified Government Program to provide reliable data link for imagery delivered from the HAB to the ground station (Government Agency + UARC).
- Machine Learning Algorithms trained on proprietary Convolutional Neural Network (CNN) model architectures to perform a variety of single and combined tasks (VC-backed startup):
    - Image / Semantic Segmentation: algorithm segments the input image into a set of categories or classes based on the model's prediction that the observed data belongs to each class.
        - Object-based Classification: identifies objects of interest within imagery then assigns them to a class or nested sub-class associated with that object. (e.g. identifies vehicle, then classifies into truck, passenger car, trailer, etc.)

        - Pixel-based Classification: algorithm assigns a classification to each pixel within the image (e.g. delineating water, roads, buildings, etc.)
        - Change Detection: detects changes in pixels across multiple images in a time series (e.g. 2017 observed pixels in area of interest = building; 2020 observed pixels in area of interest = roads).
    - Algorithmic performance was evaluated on a PR curve / based on the following metrics:
        - Precision: how often a model's prediction is truly of the correct class.
          (e.g. If a car detector model has a Precision of 0.9:
          9 out of every 10 times that the model predicts that there is a car present in an image, the car is truly present. Conversely, 1 out of every 10 predictions is incorrect.)
        - Recall: the proportion of the desired object that is correctly classified.
          (e.g. if a car detector model has a recall of 0.9:
          9 of every 10 cars that are truly present in the image are correctly found by the algorithm. However, 1 out of every 10 real cars are missed by the algorithm.)
        - As opposed to my experience with commercial applications of computer vision – which consistently considered an optimal F1 score demonstrating balance of both Precision and Recall, the defense customer's operational mission use cases required more nuanced, intentional tuning, oftentimes on a per use case or per feature class basis. (e.g. higher Recall to make sure nothing that looks like a car is missed even if the model alerts me to animals; higher Precision to only send limited personnel resources to intercept the car when the model is extremely confident that an object is present and that the object is a car.)

Due to the COVID-19 pandemic, the desired outcomes, data sources and urgency of need changed pretty drastically between the time of contract award and actual period of performance for the program. Ultimately, although originally-conceived as a system built upon a program of several commercial-off-the-shelf (COTS) technology solutions with demonstrated maturity, the program evolved into more of a green-field approach combining multiple components provided by several external and

functionally-disparate parties. Each component and point of collaboration compounded the layers of complexity to consider while confronting the administrative, programmatic and technical latency introduced into the system as a whole, as well as the trade-offs required at a detailed level that might impact technical feasibility of delivering a solution capable of achieving the program's desired outcomes.

- Data availability and access at times became a threatening blocker to computer vision engineering for a vast variety of reasons:
    - commissioning of recently-launched commercial cluster small sats providing data for our program
    - delays in the Government administering access to data, especially due to varying classification levels and security requirements for the software and hardware components being borrowed from separate programs (e.g. introducing components from a UARC-led classified program).
    - working out licensing conflicts between commercial vendors. The commercial platforms and data providers on the program were sometimes at odds with the idea of providing (even notional or passive) other commercial and traditional defense contractor entities access to their data or components.
    - working with providers to define integration and interoperability requirements necessary for making the software actually work. For example, some startup providers wanted to deliver data on a compact disc (CD) because they had not built an API. Meeting the customer's minimum latency requirements for data ingestion and computer vision processing, required a collaborative effort to define minimum viable product API calls so we could rapidly obtain not only the data in general, but data with specific qualities (captured in the metadata).
    - iteratively reconsidering use case requirements and technical feasibility based on changes to data availability. The initial training data set for some features of interest included an ontology leveraging a point-based framework for object detection tasks. As the program, available aerial platforms and imagery resolution evolved, the feasibility of detecting and characterizing features reliably within the minimum performance thresholds (evaluated on a per feature class basis) also changed. This resulted in many updates to our approach, including creating multi-step fusion algorithms that made subclass predictions based on keypoint features validated by textual databases in order to further characterize

feature classes where training data and ground truth observation were both scarce. (e.g. detect aircraft > classify as small plane > classify as cessna based on wingspan)[7]. Our new methodology required a considerable surge of new work; for example, leveraging computer vision to reliably measuring the wingspan of an aircraft required that we ditch our point-based framework, swapping it out for a Rotated Bounding Box (RBB) detection and output so that output could be mapped to a third party dataset of relevant measurements. It was also critical to create even more nuanced performance requirements – such as the minimum acceptance criteria for the third party measurement datasets to provide value within our selected computer vision post-processing workflow (e.g. minimum requirements on amount of data that must be available per feature class, density of data available, static vs actively ingested data, etc.). The collective program team had already been continuously evaluating performance trade-offs heavily anchored by the selected aerial platforms for the program – but this new approach required that we even assess trade-offs between a computer vision-only output and a computer vision-post processing output. Those trade-offs spilled into the rest of the system's tip-and-cue requirement to support decision-making; including determining whether predictions from this computer vision-post processing workflow should be used for particular use cases and circumstances vs others. Cumulatively, the impacts of this change in direction in order to assess the technical feasibility as well as increase the viability of successfully executing the program's tasks were time and resource-intensive.

- navigating nuances with Intellectual Property and Technical Data Rights. At one point I drew a diagram after I realized that we-the-program were collectively delivering a system for Government Customer A, by developing computer vision algorithms from Company B, to detect objects of interest captured over areas of interest by: data provided by and captured in Company C's platform; data loaned to us from Government Customer D, from a Program where UARC E was developing new sensor technologies, leveraging software from Program F, and for our program, we mounted that sensor to Defense Contractor G's platform to deliver the data to Government Customer H's AWS ground station, so that the data borrowed from Government Customer D's Program could undergo pre-processing and georegistration by Company I, then delivered to Company J, who trained machine learning models to detect and classify

[7] Anecdotal example; actual tasks and feature classes have been substituted.

objects and features of interest on imagery and video data, then combined that with geolocation data from Company K, L, and M, as well as RF data from Company O — to meet Government Customer A's requirements….

    - newly launched sensors or mission use cases such as detecting rarely seen objects in imagery or video. Even if strategies such as using synthetic data were available, implementing these techniques introduced additional layers of complexity due to the features of interest, varied backgrounds and terrains of the areas of interests, necessity of simulating both of these factors, as well as the pre-processing, geo-registration and normalization challenges of applying computer vision successfully across the data outputs across varying resolutions.

- Data collection and curation sometimes occurred live based on mission requirements, in turn similarly requiring model retraining to occur as close to near real time as possible (for the sake of this bullet, NRT = the latency between data capture minus data ingestion, processing, storage, and reporting / dissemination). An example included a natural disaster event where there was little data available of the area(s) of interest post-event, as well as during the launch of new sensor capabilities that had no historical data archives in which to reference and train a model.

- On multiple programs, I experienced that based on the nuances of that particular program and / or for a variety of other reasons, Government Customers / Data Providers may not want to share the input data for generating training data or for conducting model training efforts if:

    - the input data is already sensitive or classified based on its format or modality

    - the input data *becomes* sensitive or classified based on what is observed during flight

    - the sensor and information about the sensor itself (much of which is critical to understand for not only developing suitable ingestion pipelines, but also training performant computer vision models) — rather than the resultant output data — is sensitive or classified.

    - the output imagery from that system is unclassified, but some other component of the overall imagery collection platform is sensitive or classified

- the classification status of the data itself or the classification categories used by the Government changes. The most notable recent reference of the latter that impacted my programs includes implementation of Controlled Unclassified Information (CUI) as a new classification category by the Obama Administration to absorb multiple other categories of information. Formerly denoted as Law Enforcement Sensitive (LES) data (now CUI), data supplied / captured by a commercial company that sells data to both public and private sector entities, could become controlled information in the event that data becomes used for law enforcement purposes. (e.g. imagery from a UAV company that regularly images the entire city and sells that imagery to commercial developers and insurance providers happens to capture a crime and is being used to inform dispatch decisions. A subsection of the city imagery captured would become controlled data as it indicates the location(s) or likely location(s) of law enforcement personnel.)

Prepare to Operate in a Robust and Sometimes Competitive Collaboration Ecosystem

The DoD has historically endured a long yet sadly continuous failure to recruit, attract and retain personnel with subject matter expertise in engineering disciplines like artificial intelligence and machine learning. Because the DoD is competing with venture-backed startups and behemoth name-brand tech entities for the same pool of talent — not to mention dealing with additional self-imposed eligibility requirements like U.S. citizenship and security clearances — the DoD is unable to match the salary, benefits, flexibility, work culture and peer mentorship discipline-based career growth opportunities necessary to attract and retain specialists. Talent attracted by "the mission" pitch are also often able to serve government mission requirements by choosing to work for the very startups, tech companies or Not-For-Profit entities that the DoD entrusts to provide advanced technologies. So ultimately, navigating the DoD contracts entails engaging an ecosystem of semi-private vendors with "Not-For-Profit" relationships and designated government-issued acronyms (lots of alphabet soup). Although they are "Not For Profit," the extensive access afforded these semi-private entities as well as their unique claims for Intellectual Property Rights can sometimes make them seem more like startup frenemies.

- Systems Engineering and Technical Assistance (SETA) Contractors
  One approach the Government employs (pun intended) in attempts to remedy their talent retention shortcoming is by leveraging Systems Engineering and Technical Assistance (SETA) contractors. Often phrased "butts in seats"in Beltway slang, SETAs are specialist personnel sort of "on loan" to the

Government through massive professional services contracts that help the Government maintain sufficient resources and / or obtain subject matter expert (SME) for specifically identified roles. These SETA contractors sit on-site with their Government peers and for all intents and purposes, function as Government employees, though their paycheck comes from elsewhere. (It should be noted that a SETA is different from Forward Deployed Engineer that sits on the Government floor with End Users in order to help them successfully use a product from the Forward Deployed Engineer's commercial company. e.g. Palantir FDEs). Government Contractor Booz Allen Hamilton (BAH) / "Booz Allen," maintains the largest SETA presence for Government AI/ML and autonomous robotics capability development.

- University-Affiliated Research Centers (UARCs)
  UARCS are research centers at U.S. universities (public and private) that are sponsored by and maintain contractual, long term partnerships with a DoD Agency in order to develop core capabilities for that agency. Johns Hopkins University Applied Physics Laboratory ("John Hopkins APL") was the first and remains the most well-known "brand name" UARC. Supplying the defense supply chain with technical excellence, UARCs are intended to be a middle-ground by serving and rapidly responding to bespoke DoD capability needs while not displacing or competing directly against private industry / commercial companies. Via collaborative contract agreements and heavy conflict of interest compliance requirements, UARC researchers gain great access to both sensitive / classified DoD data as well as proprietary commercial company data, requirements, technology, research, prototypes, budget / program funding, and resources.

- Federally Funded Research & Development Centers (FFRDCs)
  A strategic collaboration between universities and commercial companies, FFRDCs are private-public partnerships that conduct research and development efforts to serve government capabilities that "cannot be met as effectively by existing in-house or contractor resources." Similar to UARCs and in accordance with conflict of interest compliance requirements, FFRDCs cannot compete directly against private industry for contracts. They also have government agency sponsors, including agencies outside of the DoD (unlike UARCs, which are only DoD-sponsored). Popular examples include: The Lincoln Laboratory at Massachusetts Institute of Technology (MIT Lincoln Lab) sponsored by DoD; and The Jet Propulsion Laboratory sponsored by NASA at California Institute of Technology (NASA JPL).

Get Ready for the Road Trip of Modernization's Momentous Compliance Consequences

Arguably, IT reform comprises the most robust and comprehensive legislative reform of the past 20 years, which is another reason it really excites me to be in and around this historic time of defense innovation tech. Passed in 2002, the Federal Information Security Management Act (FISMA) first underscored the importance of data and information systems to national security. The Federal Information Technology Acquisition Reform Act (FITARA) of 2014 represented Government-wide (although staggered) shift to cloud services and deprecation of costly and poorly maintained legacy systems. These "Old Guards" were traded up for evolving and newly-considered "critical" technologies such as artificial intelligence, machine learning, Internet of Things (IoT) and autonomous, unmanned systems. IT acquisition policy advanced to facilitate rapid supply of these technologies –– many of which were –– and still are –– predominantly fabricated in VC-backed commercial startups and software enterprises; some of which deliver both hardware and software systems.

However, these shiny new tech toys and their origins in commercial and VC-backed businesses are reasonably accompanied by a host of other cybersecurity, national security and intellectual property concerns. Programs such as the General Services Administration's (GSA) Federal Risk and Authorization Management Program (FedRAMP) and the recently enacted Department of Defense (DoD) Cybersecurity Maturity Model Certification (CMMC) represent temporally-relevant evolutions of the Federal Government's IT overhaul; designed to wield compliance as a shield, even starting as early as the acquisition process. These programs -- often described by startups as barriers at odds with the government's "modernization" mandate – successfully serve as prerequisites to accelerating sales, product adoption and rapid commercialization in the federal market. Business to Government (B2G) startups have to earn the trust (and not to mention, past performance history) necessary to participate in large acquisitions and programs. I've learned that although it is more of an ongoing journey rather than a destination, compliance can immensely shorten the federal sales process and timeline – in particular for SaaS solutions – once a startup has met the related compliance requirements relevant to their business, their technology, and their end users.

Part of what often makes compliance seem obstructive is the onus on each individual commercial company to figure out which compliance parameters apply to them based on their unique and evolving "environment of things." This includes layered and complex nuances for:

- Commercial businesses that sell products to government customers.

- Commercial businesses that sell products to both commercial and government customers (e.g. businesses that are not solely government contractors).
- Commercial businesses that sell hosted services / cloud based technology to government customers;
- VC-backed companies that have the Government as a customer – including limits on who the company may accept investment monies from while retaining the Government as a customer.
- and many more – depending on the startup, their technology, and their end users.

As an early employee in defense-focused teams of past product-focused technology startups, I have come to understand that the first step to a startup's successful compliance journey, therefore, is to rapidly conduct research based on its own "environment of things" in order to understand the requirements uniquely relevant to the company and its offerings. But things change. So maintaining a roadmap and backlog of these requirements (many of which can become hefty and costly technical debt) that gets groomed and updated regularly is a wise undertaking. This seems to be most successful when undertaken as an intentional, collaborative and company-wide effort. Forming a Cross-Functional Compliance Committee helps to generate steady momentum towards:

- Tracking compliance-related requirements for products, services and the company.
- Surfacing and sharing those requirements with internal stakeholders to facilitate future strategic decision-making and product development.
- Identifying specific business, product, technical decisions and needs related to compliance.
- Creating actionable plans and a roadmap to execute compliance requirements.
- Identifying resource needs associated with executing those requirements, which can be increasingly important during each funding raise (e.g. internal and external such as General Counsel review of documentation).
- Facilitating and monitoring ongoing execution (aka continuous monitoring).

A Few Closing Thoughts

Defense modernization is still happening today – and with software at the center – it seems to catch momentum in waves. Some of that momentum has morphed into the stand up of new VC-like DoD and IC-led innovation cadres, championing and sponsoring dual-use defense requirements in a deliberate and intentional commercialization of and collaboration with the startup community. Some of it has dissipated with funding challenges, changing of the guards, and the contagion of great aspirational brand name programs gone bad. It remains to be seen how this momentum will shape over time. One thing's for sure; I'm in for the ride.