



**Biometrics Development,
Security, and Operations
(BDSO)**

Solution.pdf

**REQUEST FOR
PROPOSAL**

United States
Citizenship and
Immigration Services
(USCIS)

Biometrics
Development,
Security, and
Operations (BDSO)

70SBUR19Q00000150

September 16, 2019

SUBMITTED TO

Tracey Harriot, Contract
Specialist
Tracey.B.Harriot@uscis.dhs.gov

Charlotte Edwards, Contract
Specialist
Charlotte.Edwards@uscis.dhs.gov

SUBMITTED BY

Salient Federal – SGIS, Inc.
8255 Greensboro Drive, Suite 400
McLean, Virginia 22102-3828
www.SalientCRGT.com

GSA Contract Number: GS-35F-
0609S, DUNS Number: 84-
9550983

Attn: Susanna Thornton
Phone: (703) 891-8063
Fax: (703) 891-8201
Susanna.Thornton@salientcrgt.com

This proposal includes data that shall not be disclosed outside the Government and shall not be duplicated, used, or disclosed -- in whole or in part -- for any purpose other than to evaluate this proposal. If, however, a contract is awarded to this offeror as a result of -- or in connection with -- the submission of this data, the Government shall have the right to duplicate, use, or disclose the data to the extent provided in the resulting contract. This restriction does not limit the Government's right to use information contained in this data if it is obtained from another source without restriction.

Table of Contents

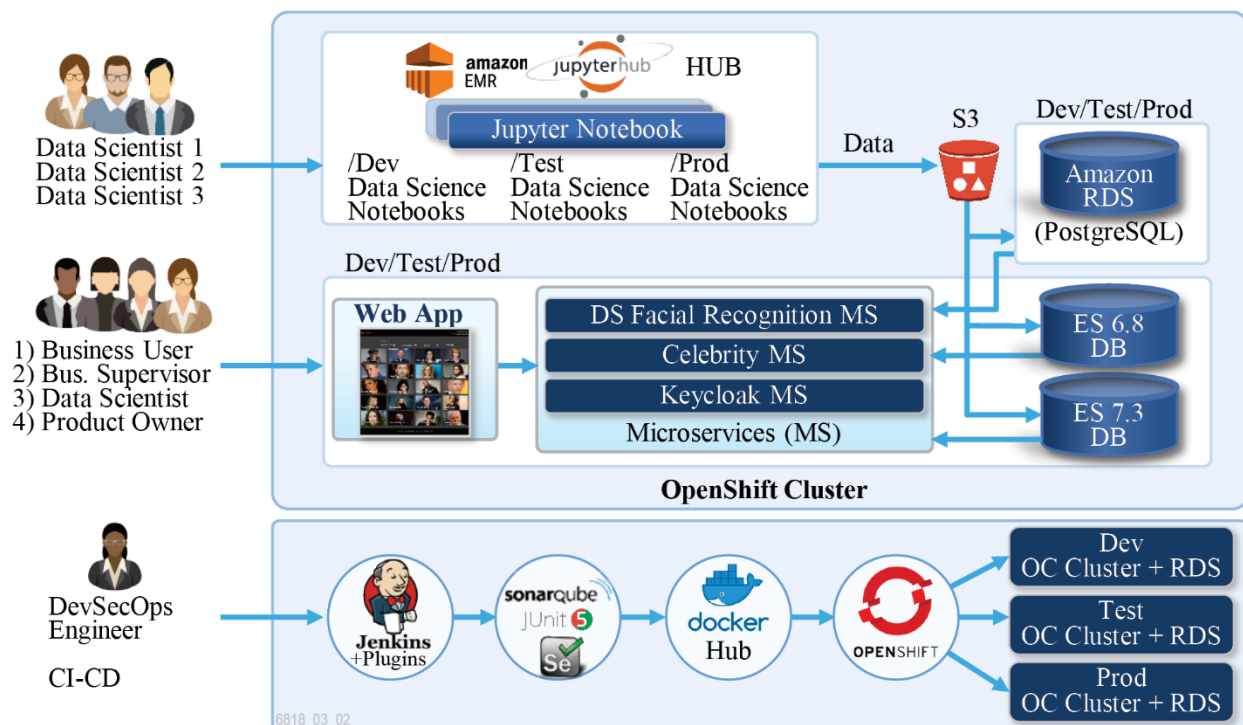
Solution.pdf (IV-9.b.1, IV-11.c.1.i, Att. 8.i, Att. 8.i.b.iv)	1
Solution Overview (IV-11.c.1.i, Att. 8.i)	1
Secure Web-Based Application (IV-11.c.1.i, Att. 8.i)	2
Microservice Architecture using 100% Automated CI/CD Pipelines (IV-11.c.1.i, Att. 8.i)	3
Interactive Notebook Runs Models on a Microservice Architecture (Att. 8.i)	3
5 Personas and 3 Solutions (Att. 8.i.b.iv).....	3
User Access	5
User Interface / User Experience (UI/UX) Methodology	5
SCRGT Code Challenge Team	6

Solution.pdf (IV-9.b.1, IV-11.c.1.i, Att. 8.i, Att. 8.i.b.iv)

Solution Overview (IV-11.c.1.i, Att. 8.i)

Salient Federal – SGIS, Inc. a wholly-owned subsidiary of Salient CRGT, Inc., hereafter is referred to as Salient CRGT or SCRGT. SCRGT planned, designed, developed, and tested a solution using solely open source tools and data, or tools from the USCIS technical landscape approved list. **Figure 1** shows our AWS full stack cloud solution architecture.

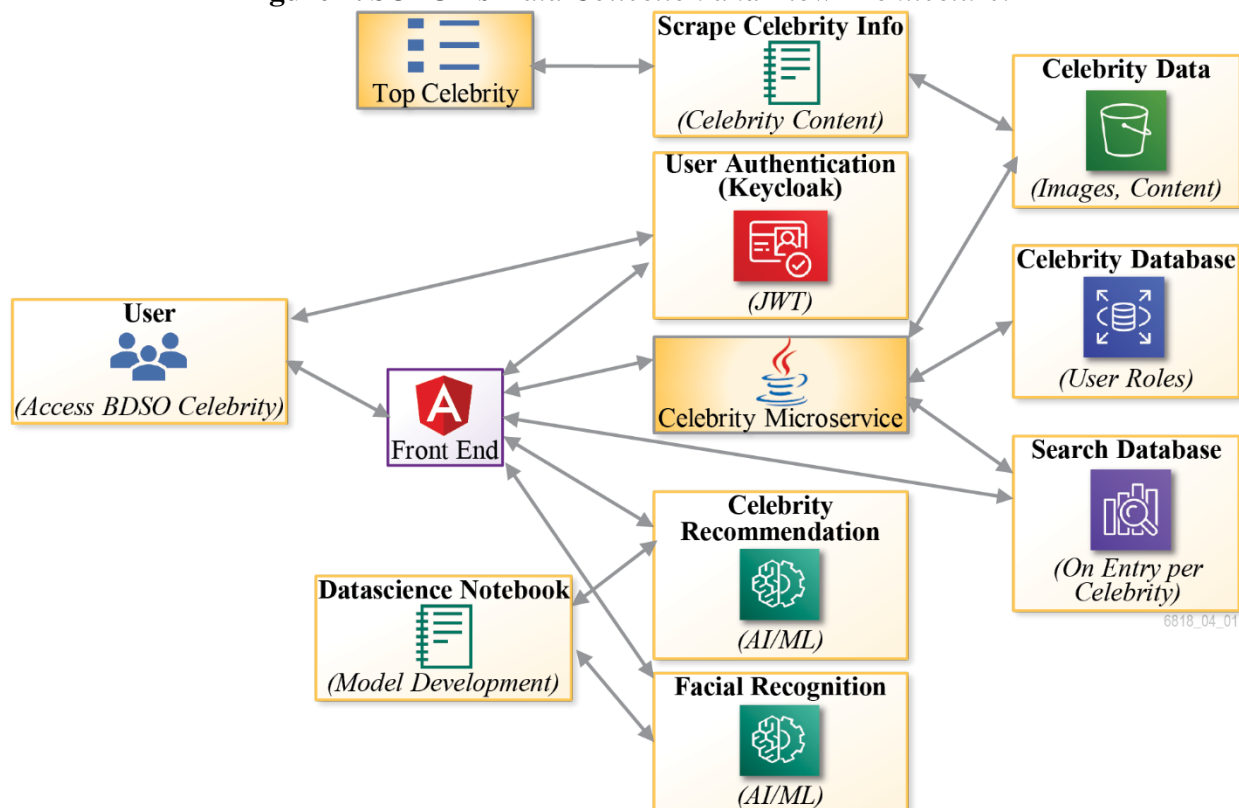
Figure 1. Our solution architecture is entirely open source or in USCIS's technical landscape.
AWS Full Stack Cloud Solution Architecture



Our data sources to provide GlobalTier Studios Top 100 celebrity information included:

- IMDb
- theMovieDB
- Wikipedia

Each of these sources provided free access to the celebrity data via APIs that populated our AWS S3 hosted data repository. **Figure 2** shows the high level workflow architecture of our data collection and process flow.

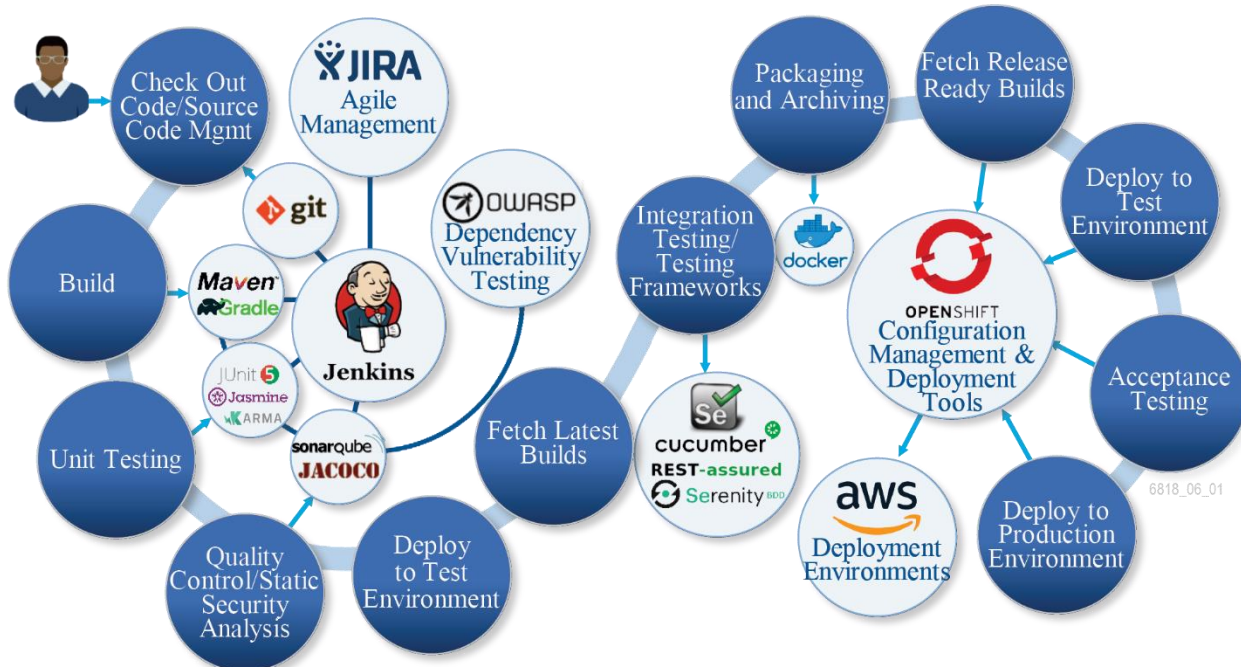
Figure 2. SCRGT's Data Collection and Flow Architecture.

Once the data was retrieved from the external sources, it was fed into our S3 bucket for storage. From there the data was called via our microservices to feed our AI/ML models, search database, and the front end. The Jupyter data science notebooks, where we developed the models, also used this data.

Secure Web-Based Application (IV-11.c.1.i, Att. 8.i)

SCRGT follows secure coding best practices in developing our solutions and applied these to ensure we developed a secure web-based application for the BioCeleb solution (IV-11.c.1.i, Att. 8.i) and validated the security of code promoted into our environments. We use SonarQube, JaCoCo, and OWASP to conduct vulnerability scans, static security scans, and code quality assessment, as shown in Error! Reference source not found., which is our 100% automated Continuous Integration/Continuous Delivery (CI/CD) pipeline. For this challenge, we integrated an open source Identity and Access Management (IAM) solution called Keycloak to provide role-based access control from the application login page. This functionality ensures that each user type can access only the appropriate application pages for their respective needs, in accordance with the core user stories.

Figure 3. SCRGT's solution includes a 100% automated CI/CD pipeline.



Microservice Architecture using 100% Automated CI/CD Pipelines (IV-11.c.1.i, Att. 8.i)

We deployed a DevOps platform via automated scripts, deploying tools like Jenkins, SonarQube, Selenium, Cucumber, Maven, JUnit, Jasmine, Karma, and Docker. We used Jenkinsfile (Infrastructure as Code) scripts hosted in GitHub for process automation. Our CI/CD pipeline is 100% automated, and deploys containerized microservices orchestration using OpenShift on AWS, and deploys to Dev, Test, and Prod environments.






Interactive Notebook Runs Models on a Microservice Architecture (Att. 8.i)

Our interactive notebooks run models and expose the resultant data to the front end BioCeleb application via microservices. We developed the models in the Jupyter notebooks with the notebooks and intermediate data produced and consumed submitted through our CI/CD pipeline. The data was presented as microservices, which are deployed into the execution environments through our OpenShift container/service orchestration platform. See [Models.md](#) for more details about our modeling, processing, training, and integration solutions.

5 Personas and 3 Solutions (Att. 8.i.b.iv)

To deepen our understanding of the business needs that our solution was to satisfy, we examined the goals, objectives, and habits of 5 personas that our user experience (UX) designer developed from the 10 core user stories. We defined a suitable, best-fit solution for each persona, shown in **Figure 4**. The product owner (PO) represents the interests of all end users – business user (BU), business supervisor (BS), data scientist (DS) – who share the PO's goals and objectives.

Figure 4. *Our UX designer developed personas to facilitate greater clarity on the user stories.*

Name, Role	Goals & Objectives	Habits	Solution
 Darren Edwards, Product Owner	<ul style="list-style-type: none"> View different classifications of top celebrity data Visualize, analyze, and see insights for multi-dimensional data types to make decisions about <ul style="list-style-type: none"> Celebrities Roles Biographic information Biometrics Recommendations Understand usage capabilities of application for all user groups 	<ul style="list-style-type: none"> Skims pages – wants to see the big picture at a glance Doesn't like "chart junk," wants ability to quickly understand visualizations Likes to cross-reference data from a variety of sources – to gain a quick understanding of connections between correlating points of information 	Web application
 Susie Blackmore, Business User	<ul style="list-style-type: none"> Use fuzzy search and combinations of information to run celebrity searches View data from multiple sources in a comparison fashion View Celebrity Profiles along with comparative recommendations 	<ul style="list-style-type: none"> Likes a search that compensates for spelling deficiencies by intuitively deducing what she's searching for Wants a dynamic search that includes filtering by Experience, Age, Genre and Gender 	Web application
 Bobbi Andrews, Business Supervisor	<ul style="list-style-type: none"> View and update celebrity data to incorporate current information and/or resolve data conflicts See comparisons in data collected by multiple sources to ascertain accuracy and reliability of information provided 	<ul style="list-style-type: none"> Works quickly with a lot of work to process Wants to search for specific celebrities using filters and fuzzy terminology Doesn't like a lot of steps to accomplish any tasks 	Web application
 Nina Kim, Data Scientist	<ul style="list-style-type: none"> View raw data on infinite scroll with infographic visualizations 	<ul style="list-style-type: none"> Reads carefully – wants to see every detail of the data Doesn't like visualizations that get too "artsy" and obstruct the data – likes clean layouts where she can see information quickly 	Web application supplemented with details contained in Jupyter notebooks
 Malcom McDowell, DevSecOps Engineer	<ul style="list-style-type: none"> Create an AWS environment, which consolidates data from multiple data sources and allows data preparation Stand up entire environment using 100% automated scripts w/minimal commands Implement AI/ML to ensure records with different spellings or context are recognized as being the same and create a single celebrity record view Maintain referential integrity when creating the view from each database Implement, demo and output results of automated security scans and code testing as part of a pipeline 	<ul style="list-style-type: none"> Absorbs complex technical information – quickly gets the gist of an issue Impatient with wordy descriptions, wants to jump to the actions Automates all functions possible to minimize manual intervention 	Scripts and instructions written for a person with technology expertise

User Access

The five personas can use one of three user types to access different pages and functionality. The Product Owner can access the entire application in order to ensure a complete understanding of end user functionality and ability to replicate usage scenarios for problem triage and new story creation. The DevSecOps Engineer does not access the application directly but administers the backend infrastructure upon which the application resides and operates.

Figure 5. Role-based user types control access to web application pages and functionality.

User Role	Accessible Resources	User Name for Login	Password
Business User	Search page	test-user	test123
Business Supervisor	Search page, edit data on celebrity profile	test-supervisor	test123
Data Scientist	Raw data infographics page	test-ds	test123

User Interface / User Experience (UI/UX) Methodology

We referenced these personas to design an intuitive and responsive web application user interface (UI) with our 75% design, 25% coding approach. As shown in **Figure 6**, our UX designer hand-sketched UI design ideas for team discussions and then created Balsamiq low fidelity and Axure high fidelity wireframes, which our business analyst, developers, data scientists, and other team members iteratively reviewed. After our front end developers received validated wireframes, they started writing Angular code to create the application front end coordinated with the design/development direction of the data scientists and their model output.

Figure 6. Wireframes allow POs to visualize the end product with minimal coding work.



SCRGT Code Challenge Team

SCRGT created our solution with a multi-dimensional, multi-discipline team, shown in **Figure 7**. Our Scrum Master/Business Analyst, Gayatri Anand Patel, facilitated team standups and other Agile ceremonies. SCRGT's Innovation Center Chief Architect of DevSecOps, Pramod Malhotra, and Senior Technical Architect, Amit Yadav, provided system architecture and overall technical leadership. Before the RFP was released, our teams conducted several code challenge preparation sessions at our Agile Readiness Center (ARC). These practices follow a playbook that we use to prepare Agile development teams for transition into new customer environments. We worked through sample user stories, learning to work in the anticipated USCIS BDSO CI/CD pipeline, and to function as a self-organizing team. When the real code challenge began, our team was immediately ready to begin analyzing the RFP as an established team with a common point of reference. During the code challenge, we used a variety of technical governance practices, derived from our experience with Agile best practices, to coordinate concurrent activities between our DevSecOps/infrastructure, data science, front end and back end application development, and testing teams.

Figure 7. Team SCRGT organized as collaborative groups focused on data science, DevSecOps, front end and back end development, and test automation.

