G2 Software Systems

G2 Software Systems, Inc. is a San Diego based small business that works primarily as a defense contractor. Formed in 1989, we have been in business for 29 years as a woman-owned small business and today we have roughly 150 employees. G2 is named for our founder, and president Georgia Griffiths, a proud CSULB alum and sponsor of several CECS scholarships.

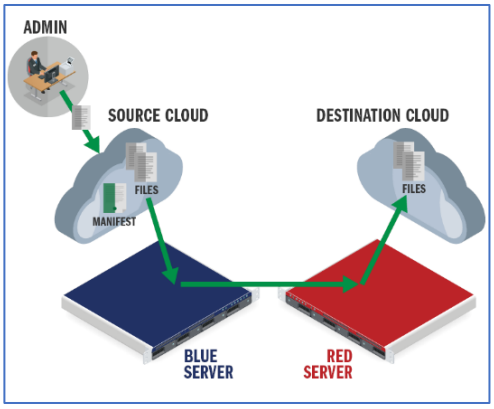

Figure 1: High Level Cross-Domain components

G2 has contracts with numerous DoD agencies, including NORAD/US Northern Command in Colorado Springs. G2 provides senior GIS data management and display services for NORAD and their many mission partners, and we have the need for a basic GIS file management web app to support a specific cross-domain file sharing requirement. "Cross-domain" here means the movement of a file from one classification to another, such as from an unclassified system to a classified one. At NORAD, G2 supports mission partners, such as National Guard unit, FEMA, state or local governments, etc., sharing custom GIS files, usually as KML, that provide amplifying information for the US DoD to provide support to those civilian authorities. Often in order to provide that support, the US DoD needs to engage their resources at the classified level, and so data provided by a state or county, for instance showing flood plain mapping data, needs to be transferred from an unclassified network to a classified one so that the DoD can engage some of their own mission systems to provide machinery, manpower, or other resources in support of civil authority.

The task is essentially to build a file hosting service that has two main interfaces. The first is a user interface component allowing human users to manually upload individual KML files (and files of other types as well, but KML is the primary interest), along with basic metadata about the file being uploaded – who is uploading it and a brief description of the file. That information, along with the file itself, is accepted by the file upload GUI, and the file and its timestamped metadata is stored in the database. The server then also exposes a machine-to-machine web service API allowing other applications to query the server for information about KML files currently available, and to pull those files as a response to other API calls.

The basic API calls the server will need to support will be provided as part of the task, as will a much more detailed list of requirements. The implementation stack, however, offers the CSULB CECS team some flexibility. For instance, the service could be implemented as a standard Java/J2EE web application hosted in Tomcat, such as with Spring for the web service API, using a basic Angular or HTML5 web UI for the front end and a MySQL or PostgreSQL database. Or, alternately, the group could choose to implement the services with something like a Node.js server component with microservices, a MongoDB data store, and Angular, React, JQuery UI, HTML5, or any other useful UI package for the basic front-end GUI.

For this project, the curb appeal of the login screen and file upload form/table are not nearly as important as clean, readable code, well-implemented server-side APIs, and if time permits, a reasonable database abstraction layer. The diagram below shows the basic flow and identifies the two main UI components (login screen and file upload/status form/table combo), as well as the server, DB, and necessary machine-to-machine web service API for use by external applications. Basic stubs can be used to simulate the 3rd party app that will invoke the team's web service APIs.

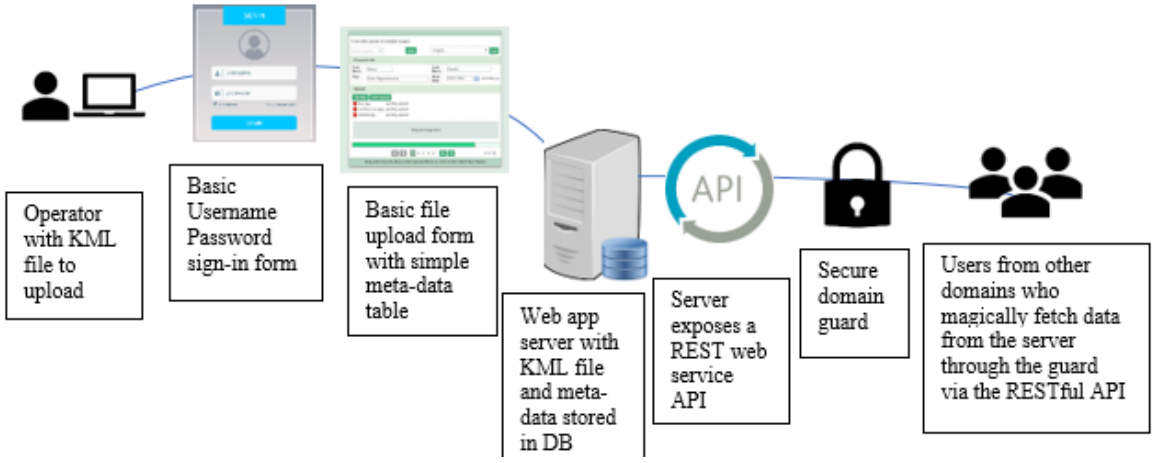

Figure 2: Basic Cross-Domain Data Flow