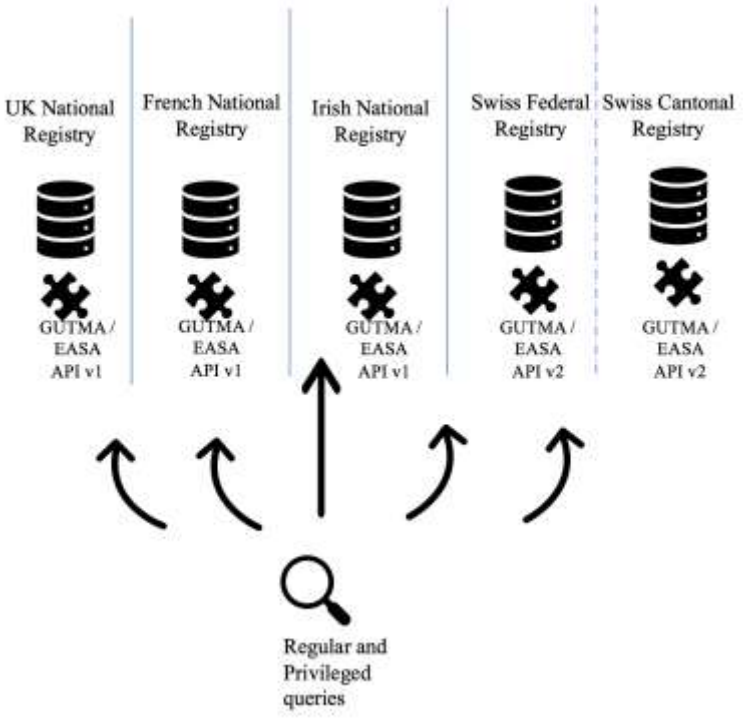
Drone Registry Brokerage

In the EU, there are regulations which mean that member states will develop a digital registry of drone operators and equipment. In many cases there will be different registries within a member state depending on the political and administrative structures in place. Each of these registries will have their own API to query it. GUTMA has been working in developing an API specification [1] that we hope that all the CAAs and EASA adopts. Adopting this specification ensures that registries developed have common language of querying its data.

The problem

We expect most queries will happen within a country, but we also expect that queries would be made across national and administrative boundaries. There are potentially hundreds of registries that we expect to be online eventually in different geographies.



The problem: Querying many different registries

For an entity trying to query registries that implements the GUTMA API specification there are many issues that they face as shown above:

1) How do they know if the registry is indeed implemented (and what version)?
2) How do they manage credentials of all the entities that they are trying to query?
3) How are changes to registry API managed? e.g. the GUTMA API specification is at v1 but in the future when it develops and v2 is released, a CAA might implement v2, how does a CAA know which API is available and queryable.
4) How does one CAA know the URL of the registry service of another CAA?
5) How does a registry know if the entity querying it is indeed privileged? i.e. police

These are not problems of an individual registry, any registry operator will face these issues and will have to develop solutions independently.

Use Case

A CAA or interested party in Geneva wants to query the French registry (neighboring country), given the proximity of France and Switzerland, the Swiss CAA might just make a special agreement with the French CAA and have logins and credentials for their registry. In such a case, there is no need for the Swiss CAA to use the broker. However, if the CAA wants to query the registry in Estonia for e.g., they may not have contacts or credentials to query it in such a case they might use the broker. What this does for the CAAs is gives them flexibility to maintain individual relationships and not have to worry about developing a relationship with all 27-member states (and beyond). This also means that the Swiss CAA does not have to store credentials for different registries, maintain them etc.

How to address these problems

There are many ways to address this problem. Let us begin with non-technical solutions:

1) Hold a monthly or quarterly meeting between the CAAs
2) Develop a portal to share the latest in updates and between different entities. Allow the CAAs to engage and announce and clarify changes

While these methods are useful and can work, we acknowledge that most of the CAAs will outsource the actual registry work to their IT vendors or external third parties who might not be interested or motivated to engage. A software solution to this problem is to build a bridge between the individual registries and the clients / interested parties querying it. Conceptually, it would look like the following:

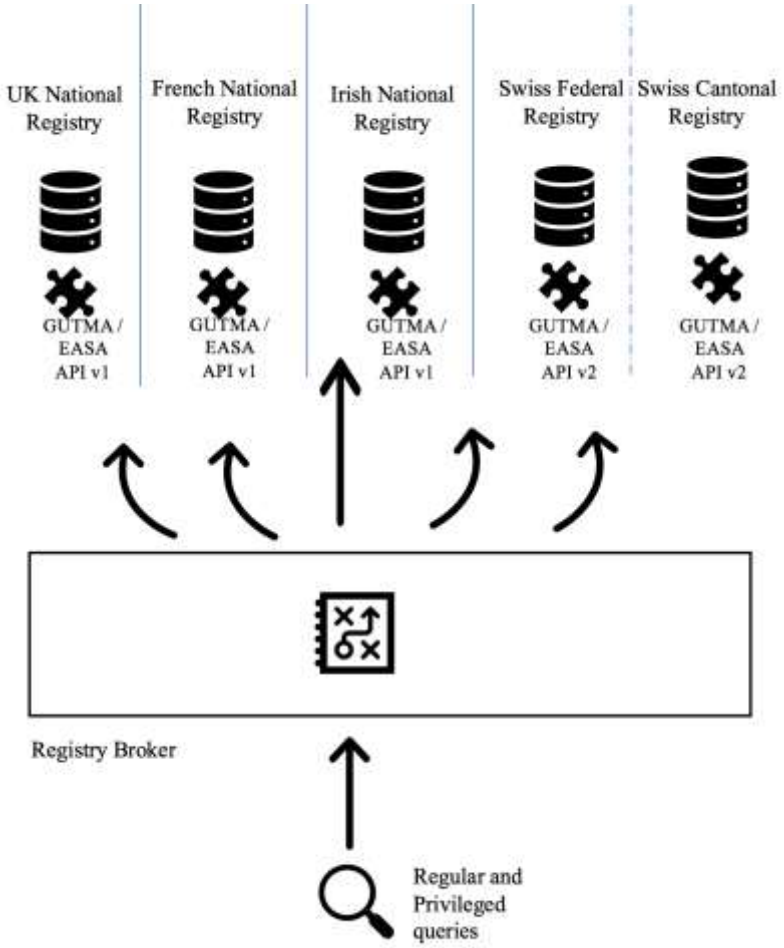


Figure 2: Decentralized Querying vs Centralized Querying

There are many advantages to building a system like this that reduces the engineering cost and provides certainty for organizations when querying different registries. It ensures that the Broker / Brokerage takes care of the changes in the individual registries.

In this case the interested party must implement just one API to the broker to ensure that queries would work without having to worry about implementing different API calls. As a broker the organization that runs it has an interest in working with the different CAAs to implement a common API to ensure that the re-work associated with the software is minimum. By having a central broker, the originating party consolidates the requests and has leverage over the registry operators

Pros and cons

Pros	Cons
- Individual Interested parties will not have to implement many calls to the registry - A standard way to communicate and query different registries (like the VAT system) in EU, ICAO aircraft registry	- If there is a brokerage, what is the incentive for individual registries to follow the GUTMA API - This is a level of abstraction that might be unnecessary

- A single point of contact for querying information like FAA - A third party maintains the registry interoperability and provides the industry and other actors a standard way to query and access this data.	- The different entities running the registry might not talk to each other (they should) - Data storage vs. querying does the broker store the data or just query it? - Security and confidentiality of the information is critical and the best way to ensure that no confidential data is compromised is by not storing it permanently.

Authentication

The primary goal of this spec is to develop a brokerage layer where incoming requests are routed to the individual registry and then response queries are then routed to the registry operator. Following are the authentication credentials required for the parties in this ecosystem.

Interested Party	Broker	Registry Operator

- Authentication credentials for regular queries
  o Account creation
  o Email contact address
  OR
- Throttle all unprivileged requests
 	
- Privileged authentication credentials e.g. police, military etc.
  o Some certificate that the party is indeed a privileged entity
  o Authorization keys for that entity (open question: how keys are recycled)
	- IF unprivileged entities are throttled then no need for any authentication information for regular interested parties, only the destination county and the type of information (API call) that is being made.
	
	- Maintain a list of privileged client access tokens (and documentation)
	
	- Maintain list of documentation and certificates etc.
		- Maintain username and password for operators and any other interested party that wants to query the registry.
		- Maintain separate documentation and credentials for privileged access.
		- Maintain credentials for Brokers like GUTMA to ensure that access is maintained

Notes on privileged access

Given the complications of maintaining documentation and keys for privileged access, some ideas that could be pursued:

- Privileged access should be made directly to the individual registry
- Registry operators have access to a share with all the certification information and details as a shared library / document store.
- In the case where privileged access requires querying many servers to generate a “merged” view of the information, this may need multiple queries.

- For the first version, it maybe useful to not have privileged access as a part of the brokerage but have it as a feature for a later date

Servers

This type of brokerage service will be hosted on servers in Europe and managed out of European data centers. The requirement would standard and in compliance with EU and other government data protection regulations.

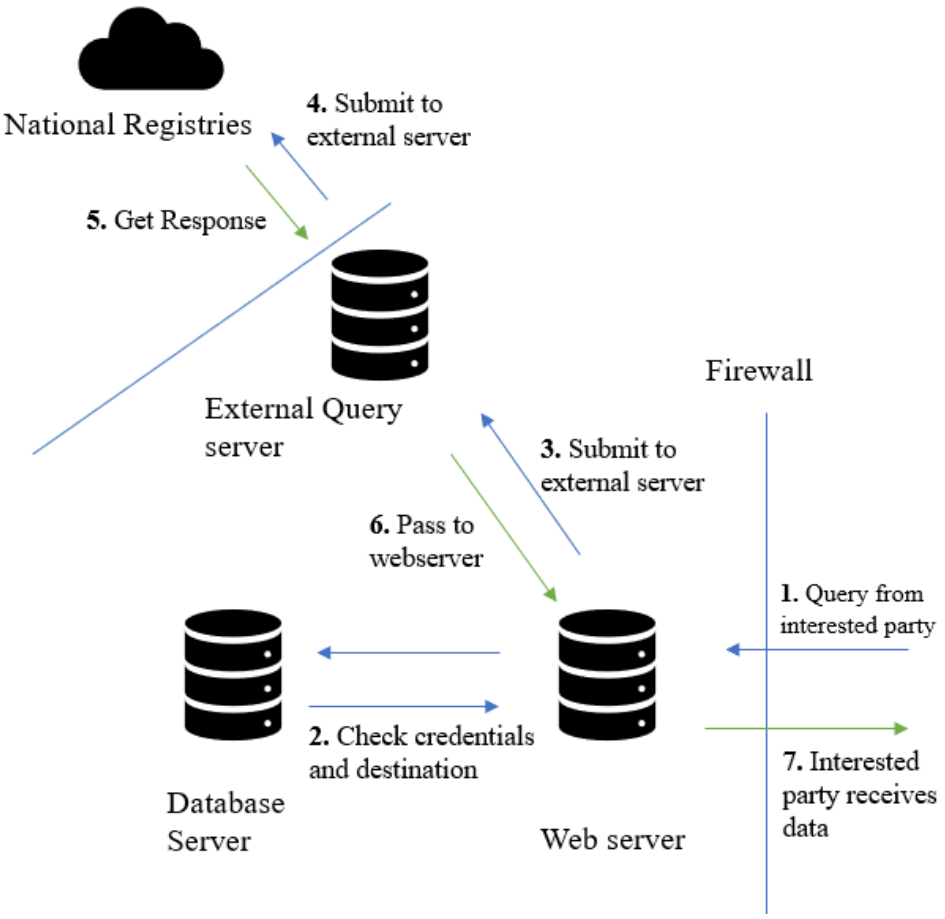


Figure 3: Sample flow of queries and traffic

In this setup, no data is stored on the broker servers other than the credentials for queries to the national registries. The interested party makes a query to the broker who in turn passes the query to the appropriate registry.

Security

For most registry queries, security considerations are straight forward, in that it is a routine query with standard credentials and no sensitive information is transferred. Using standard internet-based security of web tokens should suffice in this case. Proper firewalling and management of data base is necessary.

Conceptual Architecture

This section details some of the architectures that can be pursued for this problem from the least data requirement to maximum data requirements for the solution. And a pros / cons analysis of each approach is conducted as well. An analysis based on data requirements is important because important confidential data is potentially passed through the system.

A. Apache Zookeeper

A useful approach in this context is if the GUTMA API is successfully adopted, then to prevent duplication and re-architecting the API calls, a simple implementation of Zoo keeper would suffice where the individual registries are managed by the individual CAAs.

B. API Broker

A API broker architecture as described above, needs to store credential information but the actual data is stored in the database owned by the CAAs outside the system.

C. Data Scraping and storage

A alternative architecture to overcome the issue of data and external queries, is to scrape and have a dump of data and the broker stores it. This is a case where data is scraped from the registry and stored in a database owned by the broker.

A. Zookeeper		B. API Broker		C. Data Scraping and Storage	
Pros	Cons	Pros	Cons	Pros	Cons
- No data storage - Simplest to implement - Minimal software development - Safest for Brokerage operator.	- All actors must implement GUTMA API - No control over performance, if an API call breaks - Ecosystem not ready for a "hands-off" approach.	- Flexible architecture - No data storage - Simple architecture - Least amount of data storage	- Need to maintain credentials - Response times cannot be guaranteed - Duplication of query procedures + data - How handle privileged queries and responses?	- Fastest performance and query response - Most control for operator - Management of changes / updates	- Data may not be "fresh" - Rigorous controls and data safety procedures - GDPR issues - Security and other clearances.

References

[1] – https://droneregistry.herokuapp.com

Version History

14-March-2019	V2	Added Comments and review from NUAIR	Dr. Hrishikesh Ballal / Andy Thurling
23-Sep-2018	v1	Initial Draft	Dr. Hrishikesh Ballal