Reservoir sampling as a service

Wikipedia describes reservoir sampling as:

> "... a family of randomized algorithms for randomly choosing a sample of k items from a list S containing n items, where n is either a very large or unknown number. Typically n is large enough that the list doesn't fit into main memory."

https://en.wikipedia.org/wiki/Reservoir_sampling

The goal of this challenge is to implement "Algorithm R" from the above Wikipedia article and wrap that algorithm in a web service that can be queried via curl. The service API must support the following operations:

- Begin a session by uploading a list (L) containing K starting integers
  - You can assume that K is a small number such that this list (L) fits in memory
  - The size (K) of the list may vary from session to session
- Upload a new integer to a given session
  - This integer has a chance to displace a random integer in the session's list (L)
  - Algorithm R explains how to compute the probability of displacing an element
- Close the session, returning the current list (L) of K integers
- This service must support multiple concurrent sessions

For example, if the user:

- Begins a session with 3 integers: A, B, and C
- … then uploads a new integer D
- … then uploads a new integer E
- … then closes the session

… then they should get back a list of 3 integers where each integer is equally likely to be one of A, B, C, D, or E without duplicates.

If you implement Algorithm R correctly then:

- each session should have O(K) space complexity
- uploading an integer should have O(1) time complexity
- every integer you upload should be equally likely to be present in the final list

The submission must be a self-contained project with instructions on how to build, run and query the service. You can complete this challenge in any programming language of your choice