Prior Designs and Implementationsch:previous-designs

Before working on ShareTrace, I did not have experience developing distributed algorithms. The approach proposed in ch:proposed-design is my fifth attempt at defining a performant implementation of risk propagation that is also decentralized and online. The prior four attempts offered valuable learnings that guided me toward the proposed approach; however, only the latter supports truly decentralized, privacy-preserving contact tracing. To document my efforts in developing this thesis, prior designs and implementations are provided in this appendix.

Thinking Like a Vertexsec:giraph

The first iteration of risk propagationhttps://github.com/cwru-xlab/sharetrace-giraphtilized Apache Giraphhttps://giraph.apache.org an open-source version of the iterative graph-processing library, Pregel Malewicz2010, which is based on the bulk synchronous parallel model of distributed computing Valiant1990. Giraph follows the "think like a vertex" paradigm in which the algorithm is specified in terms of the local information available to a graph McCune2015.

Risk propagation was implemented as defined by Ayday2020,Ayday2021, using the factor graph representation of the contact network. Moreover, the implementation assumed the use of Dataswyft Personal Data Accountshttps://www.dataswyft.io which provide a data-oriented interface to self-sovereign identity [pp. 98–99]Preukschat2021. However, because the Exposure Notification API developed by Applehttps://covid19.apple.com/contacttracingnd Googlehttps://www.google.com/covid19/exposurenotificationsoes not permit remotely persisting ephemeral identifiers, the implementation assumed that an individual's geolocation data would be analyzed to generate the factor in the factor graph (sec:contact-search). fig:aws-architecture describes the high-level architecture. Callouts 1, 2, and 4 were implemented using a fan-out design in which a ventilator Lambda function divides the work amongst worker Lambda functions.

figure[htbp] [width=]aws-architecture [ShareTrace batch-processing architecture]ShareTrace batch-processing architecture. 202 An AWS Lambda function retrieves the recent risk scores and location data from the Dataswyft Personal Data Accounts (PDAs) of ShareTrace users. Risk scores are formatted as Giraph and stored in an Amazon Simple Storage Service (S3) bucket. Location data is stored in a separate S3 bucket. 203 A Lambda function performs a contact search over the location data and stores the contacts as Giraph edges in the same bucket that stores the Giraph . 204 Amazon Elastic MapReduce (EMR) runs risk propagation as a Giraph job and stores the exposure scores in an S3 bucket. 205 A Lambda function stores the exposure score of each user in their respective PDA. fig:aws-architecture

.b/sharetrace-giraph u
//giraph.apache.org ,
//www.dataswyft.io ,
.com/contacttracing a
xposurenotifications d