Computing Collective Knowledge from User-Siloed Data in Decentralised Applications

Mosaic Runtime

draft - last revision May 10, 2020

Project Vision April 2020

Abstract

We construct a general framework of smart contracts on Ethereum that can be used to build decentralised applications which provide any service hosted from a decentralised network of application nodes. We move the application code and content to an encryption-derived access-control DAG building on IPFS and ThreadsDB. The contracts allow for orchestration of redundant computation and services to build out the application. We present contracts for orchestrating high performance computation, again over a decentralised network of nodes, allowing for deriving rich analysis and feedback loops. Applications can also build on the computations of other applications, extending the driving strength of the world-wide-web, then smart contracts now into high-performance computing.

We present a special type of such application to provide an extension of BFT consensus over Ethereum smart contract state, such that the contract orchestration is feasible in scale and cost. Lastly we elaborate on a construction for managing different application data together under a single root for the user to manage their user-siloed data and enable secure, decentralised and easy-to-use account recovery.

I. Responding to the COVID-19 pandemic

The COVID-19 pandemic has caught the world underprepared. Contact tracing can be a valuable tool to identify who to test for an infectious disease, and as such control its spread - COVID-19 acutely so.

The fight against epidemics has altered the course of social and political history. In particular the creation of public health systems (and governments to support them) grew out of the people's acceptance for disease control as a public responsibility.

To scale contact tracing today various governments are turning to technology. Mass-collected cellphone geolocation data is used track the movement of aggregate populations, or in some countries to track individual citizens meeting each other. New initiatives are worked on to establish contact tracing with better privacy protection - we highlight DP-3T[1] - where Bluetooth signals are used to register proximity of people.

At a minimum a discussion is warranted about which technical freedoms we value, and which options are available to work on. With this memo, we look to contribute to such a discussion.

i. Introduction

Contact tracing applications elevate the question of who owns such intimate user data. Conventionally data is owned and accessible by the application developers. In contrast a growing body of open-source protocols is building towards applications with *user-siloed data*: where only the user owns and can access their data.

However, during the current pandemic, the

major technology platforms and governments are accelerating the trend to exploit user data, first to fight the pandemic and misinformation, but it is an open question whether such new powers will have oversight and be constraint for other purposes.

Therefore a contact tracing application is an ideal case study for advancing applications with user-siloed data, where cryptography enforces digital freedoms and ownership of our data. However, a contact tracing application highlights a second requirement: to meaningfully address global problems - such as the spread of infectious diseases - an application must also be able to *compute collective knowledge* from (anonymous) user-data contributions: e.g. "*What was my exposure risk to COVID-19 given later attestations of positive individuals?*".

In this document we explore how to build such applications. We bring together existing protocols, and introduce new work to construct an operating system for applications where users can collectively build insight from their data. We describe how the MapReduce algorithm can be orchestrated over a decentralised network to compute over a large user-owned data flow. All of the ideas presented in this work are being worked on under open-source licenses and this is an invitation to discuss and question, and ultimately help strengthen protocols towards a new standard for applications.

II. The Mosaic Protocol

i. Root, Hosts and Avatars

From the user's perspective Mosaic-OS is an *access-control schema* to manage their applications and data. We set forward three access levels *root*, *host*, *avatar*

Notes

[1] Decentralized Privacy-Preserving Proximity Tracing, github.com/DP-3T

References