Framework for **Peer-to-Peer** **Data Sharing** over **Web Browsers**

Vishwajeet Pattanaik

Tallinn University of Technology

Estonia

Motivation

- the Web has become increasingly **centralized**
- ever **rising** numbers of **data thefts**
- **data silos** being used to not only **harvest** user **data** …

… but, also to **manipulate user** mindset, and to **spread fake news** and propaganda

Related Work - I

- **Musubi** – A Mobile Social Network and App Platform [2012]
  enables users to share data in real-time feeds.
  - supports end-to-end public key encryption
  - allows users to store app data on phones (i.e., locally)
- Gaps:
  - doesn't have support for public data sharing
  - relies on a server for message transfer

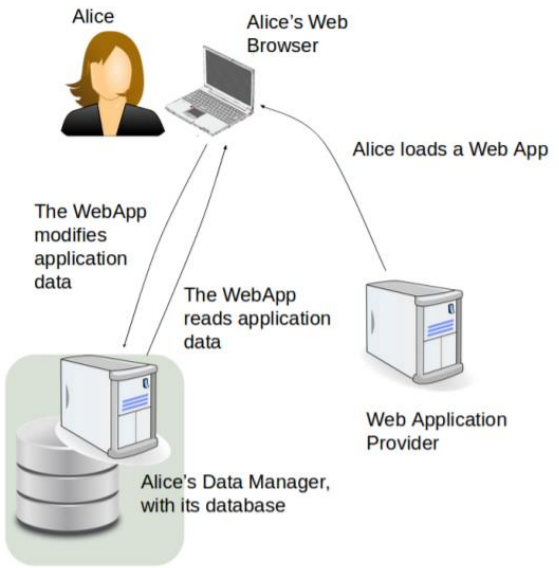
Related Work - II

- **CIMBA** – Client Integrated MicroBlogging Architecture [2014]

  is a decentralized social web application that enables data ownership by allowing users to choose where their data is stored.

  - fully decouples the application web server from the user's db
  - supports data reuse (allows apps to reuse user's social graph)

  - *superseded by SOLID*



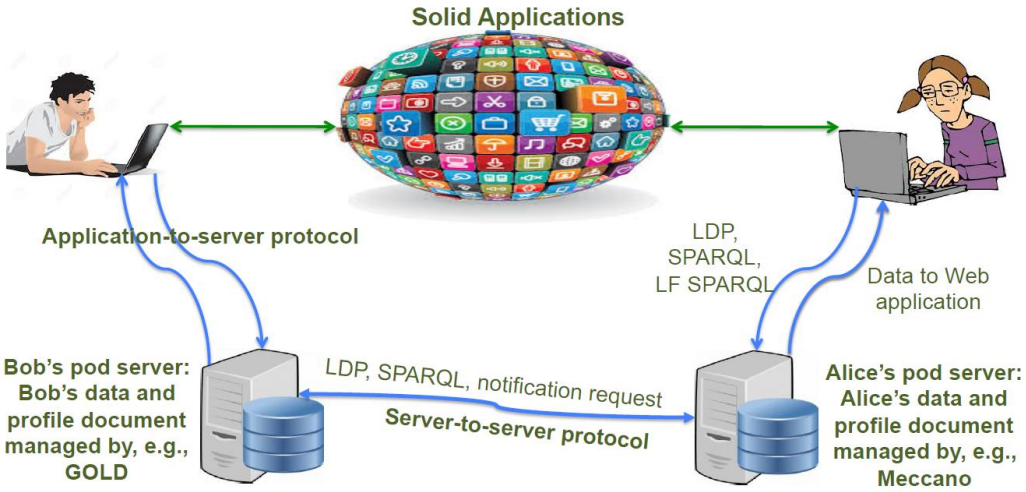
Related Work - III

- **SOLID** – social linked data [2016]

  is a decentralized platform for social web applications, that allows users to manage their data independent of web applications.

  - users are required to store their data in personal ‘pods’.
  - users have full control over how their data is accessed.

- Gaps:
  - tasks like setting up servers or finding (pod) hosting services might be difficult for lay people



Related Work - IV

- **Dokieli** – [2017] is a client side editor for decentralized article publishing, annotations and social interactions.
  - supports social interactions and allows users to retain the ownership of their data.
  - documents are independent and interoperable
- Gaps:
  - although RDFs have been around for two decades, there are still some obstacles for its broad acceptance

Challenges

- How to establish P2P network between web browsers?
  - requires investigation into Socket.IO and WebRTC
- How to create a server-less P2P network?
- How to forward messages to a user who is currently offline?

Functional Requirements

- private data sharing
- public data sharing
- private data sharing between friends (i.e., within communities)
- dealing with offline peers
- storing the data
- security & integrity
- developing a browser based solution
- *‘dealing with connection establishment, with hosts behind NAT’*

Proposed Framework

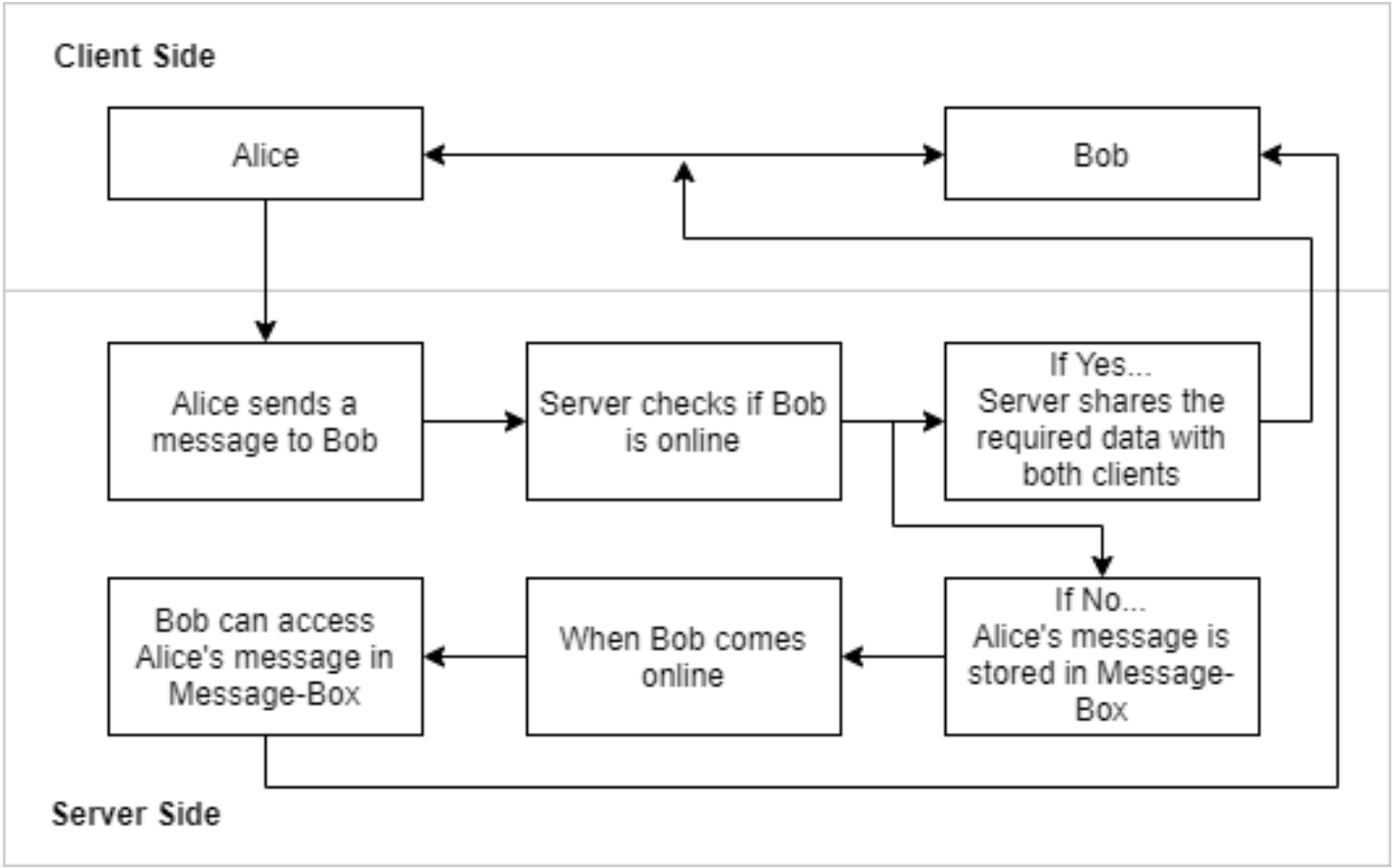
TAL TECH
DEPARTMENT OF SOFTWARE SCIENCE

- P2P connection establishment using a ***'server'*** [ONLY]
- Public data holding
- Sharing data when peer is offline

Artifact

- Live – Rooms
- Message Box
- Client – Side
- and some APIs… like: sync(), publish(), saveData()…

try the code →



Summary

TAL TECH
DEPARTMENT OF SOFTWARE SCIENCE

- Motivation
  - rising issues of data ownership and privacy on the web
- Approach
  - server-less message exchange [in best case]
- Results
  - the framework is undoubtedly useful for applications, where real-time data sending via peer-to-peer networks is critical

What next?

TAL TECH
DEPARTMENT OF SOFTWARE SCIENCE

- try to ***re-decentralize the Web***

- How?
  - building **crowdsourcing information system**

  … designed as **social platforms**

  … enabling **data critiquing** and **content quality monitoring**

  … ***without relying on developers***

“I think people’s fear of bad things happening on the internet is becoming, justifiably, greater and greater…”

“…If we leave the web as it is, there’s a very large number of things that will go wrong. We could end up with a digital dystopia if we don’t turn things around. It’s not that we need a 10-year plan for the web, we need to turn the web around now.”

- Tim Berners-Lee, the inventor of the web

@ launch of “Contract for the Web”

Questions?