ECE:5550:0001 Spr18 Internet of Things Project: SenseAd

Bakir Hajdarevic[1], Daniel Machlab[3], Ryan Wedoff[2], and Luka Zdilar[3]
[1] Department of Biomedical Engineering, [2] Department of Computer Science
[3] Department of Electrical and Computer Engineering,
University of Iowa, Iowa City, IA

Abstract

We have developed a novel advertisement platform that issues payments to users using the next generation public distributed ledger, IOTA. A user recommendation system based on user and sensor data determines the appropriate ads for users to view. The platform that we have developed serves as a prototype for a smart city infrastructure which will inevitably incorporate other smart city analytics. In our current prototype, kiosks are set up and distributed throughout a city. Users must register into the system via a mobile app where they set up their IOTA receive address and provide a full-face photo. Once stored in a large database, a kiosk may then identify users using facial recognition services provided by Microsoft Cognitive Services. The quantity of IOTA received are determined by the length of time and number of advertisements are viewed.

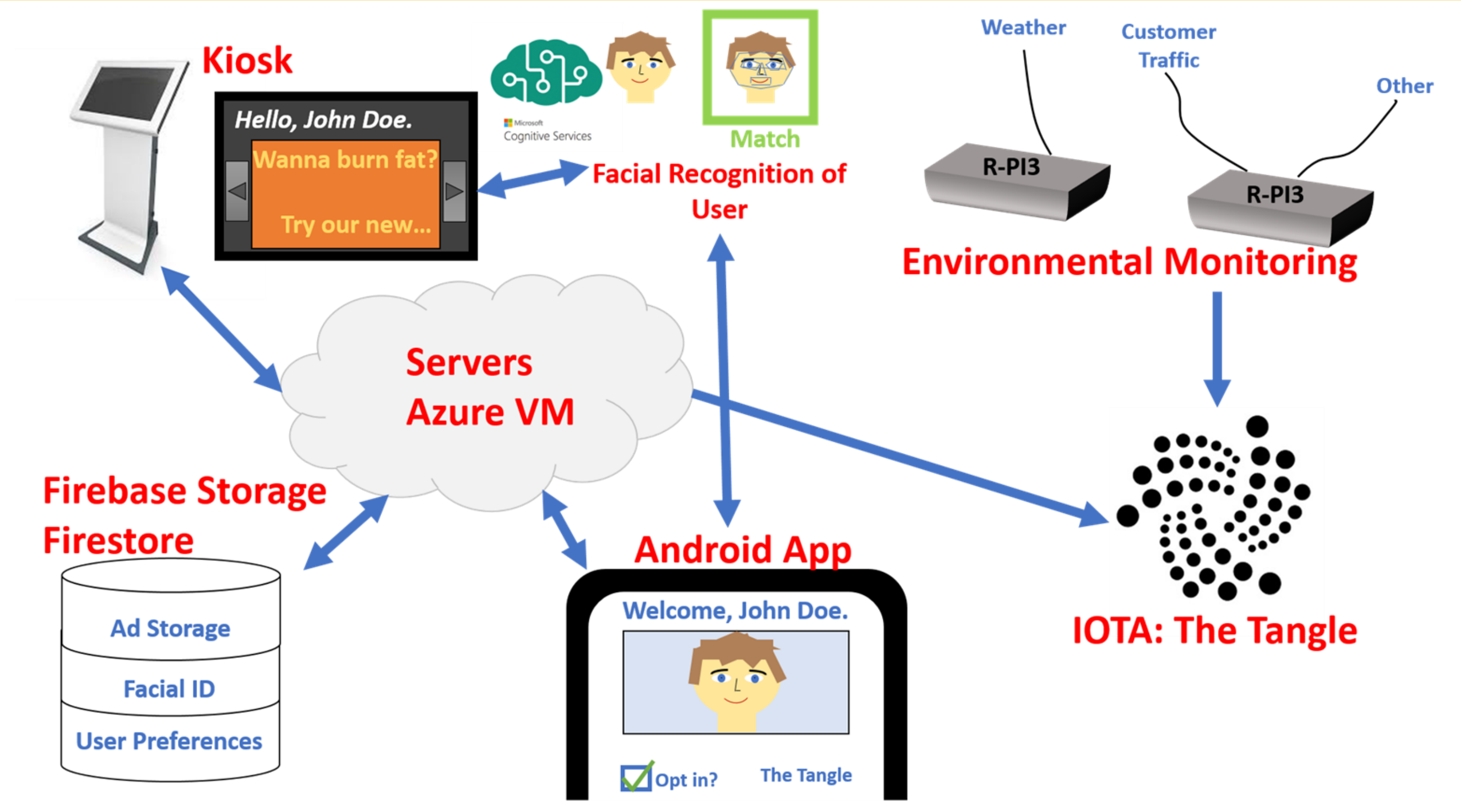

Figure 1. Overview of SenseAd System Architecture. The system consists of 6 essential subsystems: android app, environmental monitoring, Google's Firebase, IOTA, and Servers, specifically Microsoft's Azure VM. In our current model, kiosks are set up and distributed throughout a city. Users register into the system via a mobile app where they set up their IOTA receive address and provide a full-face photo. Kiosks may then identify users using facial recognition services provided by Microsoft Cognitive Services. A user receives IOTA based on a short length of time and number of advertisements viewed.

The Tech

Table 1. Table of all the subsystems, the tools used during development, and our rationale for their inclusion. The primary programming languages were Java, Javascript, and Python. Firebase was used for storing advertisements and weather information. Facial recognition software was implemented using one of our team member's subscription to Microsoft Cognitive Services. Weather information was obtained using a Raspberry Pi 3 Sense Hat.

Subsystem	Tools	Rationale
Kiosk	Python, Kivy, Camera, Microsoft Cognitive Services Facial Recognition API, RPi3	Kivy GUI and frontend, kiosk running on RPi3, MCS Facial API for facial recognition
Mobile App	Native Android (Java), Microsoft Cognitive Services, Firestore, Google Play Services (QR code)	Native used to use more in-depth apis, MCS Facial API to add faces, Firestore for saving person information
IOTA Integration	JavaScript, IOTA Payments API, Azure Ubuntu VM, IOTA MAM	IOTA Provides JS APIs, Azure VM hosts analytics code and apis
Recommendation System	Python, Surprise Recommender, SKLearn, Azure Ubuntu VM	Various libraries to run the reco. system. Azure VM runs code in background
Database	Firestore, Firebase Storage	Firestore provides flexible and easy to use storage. Firebase Storage used to store image ad. files.
Environmental Sensing	RPi3 Sense Hat, NodeJS, PIR Sensor	Integrates easily with IOTA MAM

System Overview

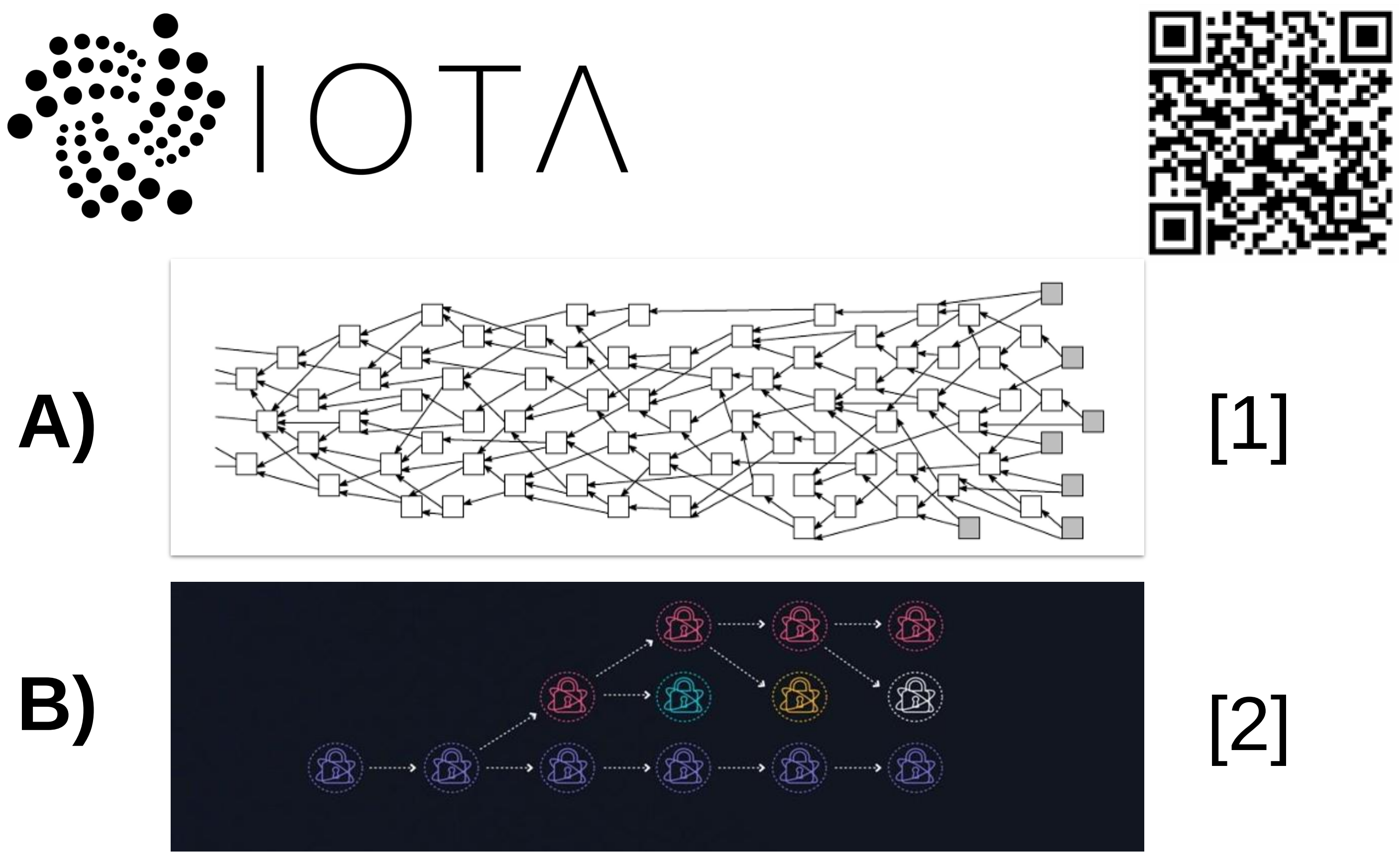

Figure 2. Overview of IOTA and The Tangle's usage. IOTA is a distributed ledger protocol built on a DAG data structure known as The Tangle illustrated in (A). Users are paid in micropayments of IOTA: A fast and feeless cryptocurrency. (B) Sensor data is streamed to The Tangle for data integrity and availability achieved via merkle tree signature schemes. The QR code shows simulations and illustrations on how The Tangle works.

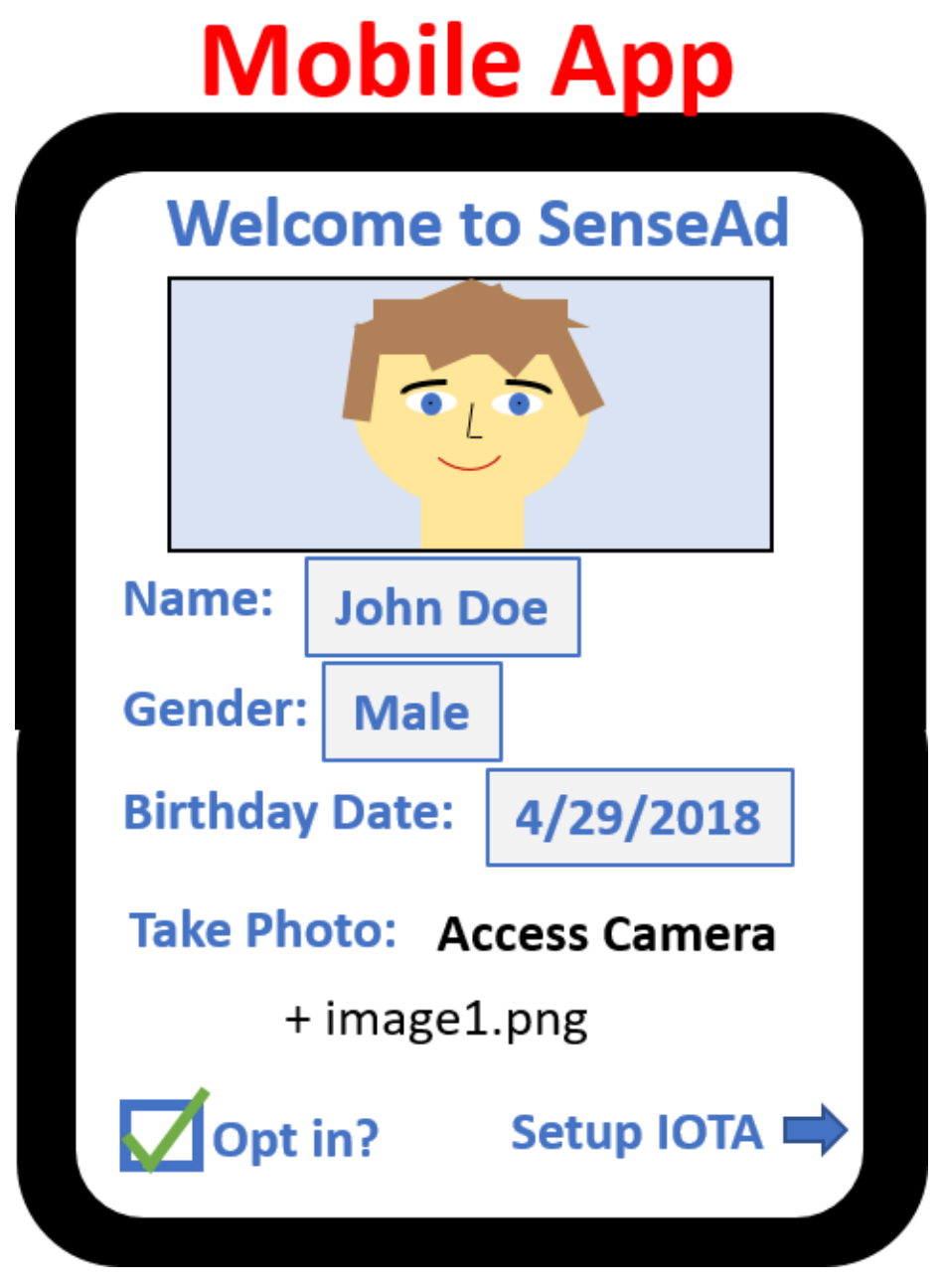

Figure 3. Illustration of the essential components used to register a user in our mobile app. A user provides basic personal information and agrees to opt into the system. In addition, a user must provide a full-face photo which is stored in a large database of existing users. Lastly, they must also set up their IOTA receive address to receive their IOTA.

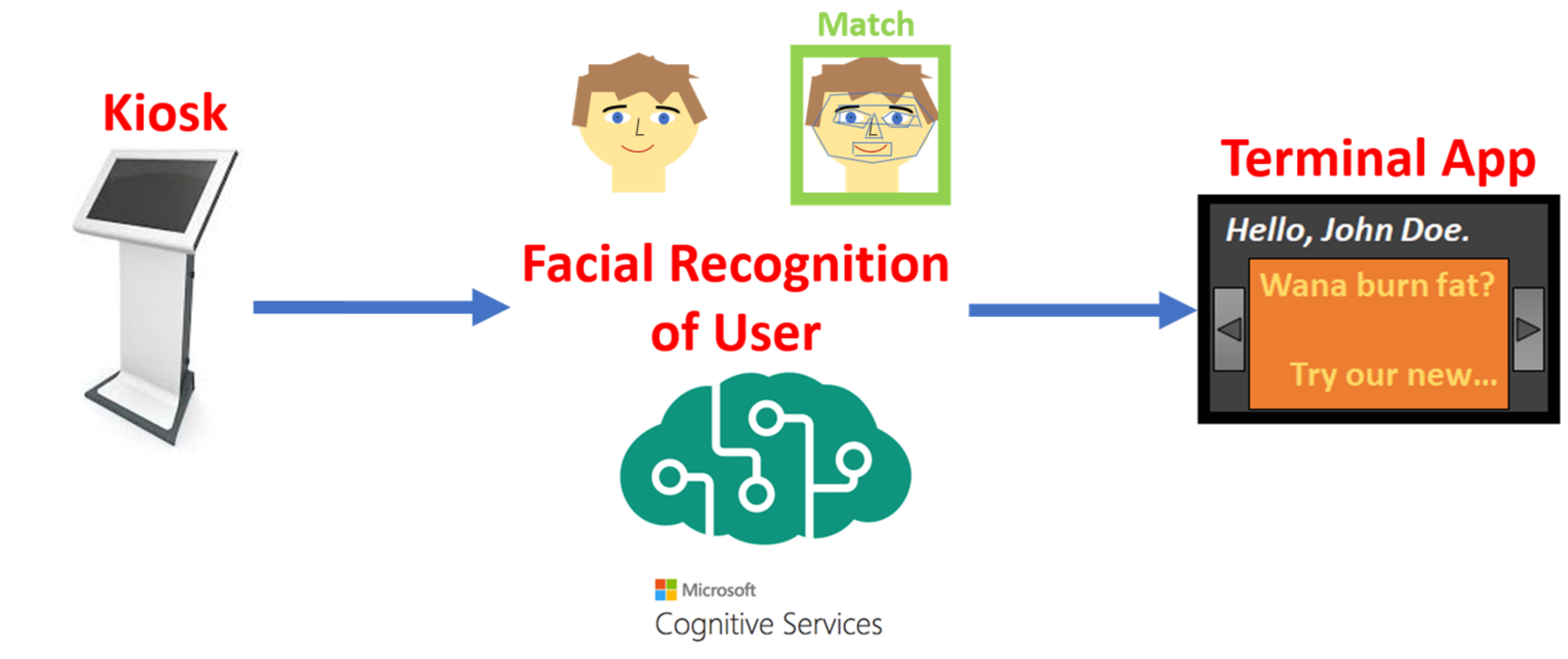

Figure 4. Diagram of how a kiosk is used. A user is identified using facial recognition software provided by Microsoft Cognitive Services. Ads are stored in Google's Firebase whereupon our recommendation system determines which advertisements to show per user.

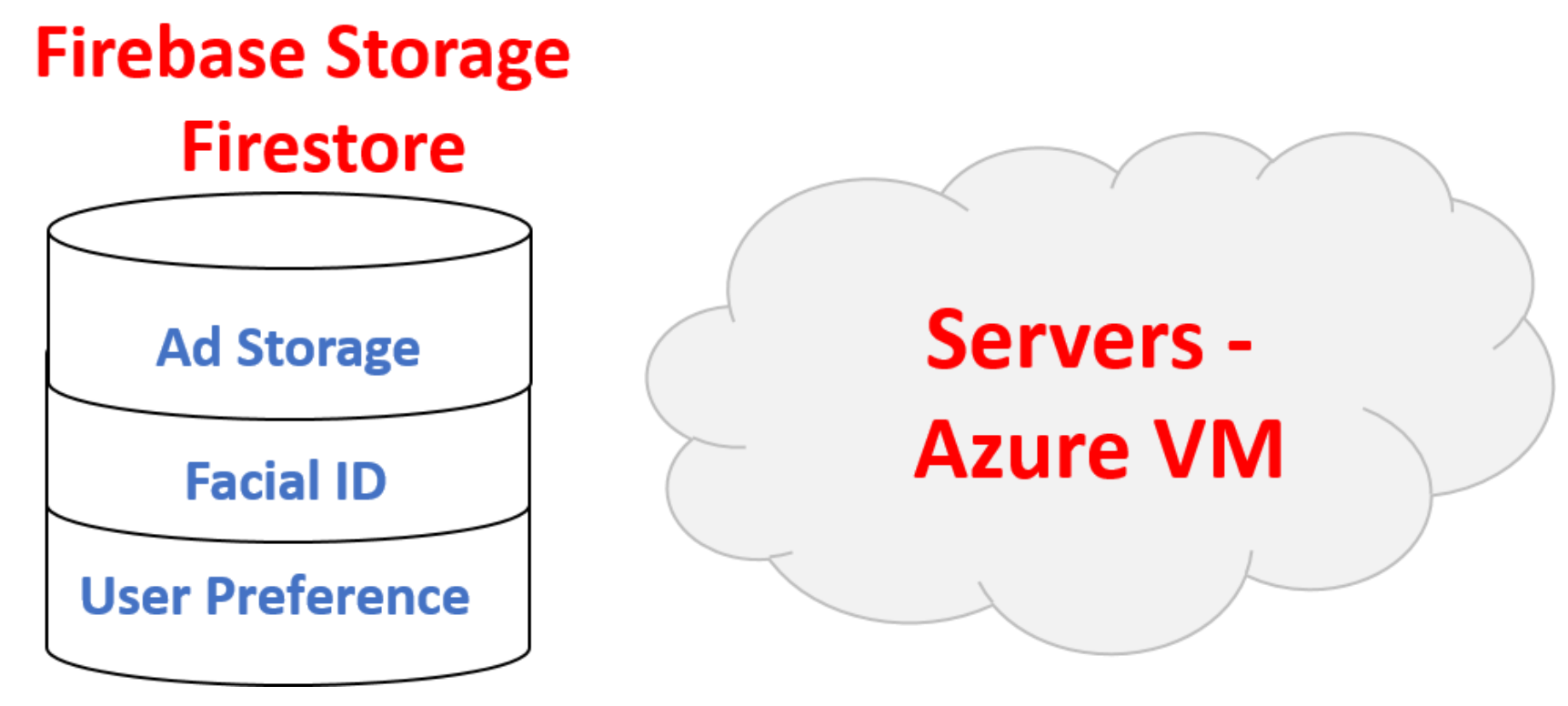

Figure 5. Illustration of the database and storage system used for storing advertisements, user facial identification, user personal preferences, and sensor information (not shown). Firebase was used for storing data and Azure VM for distribution

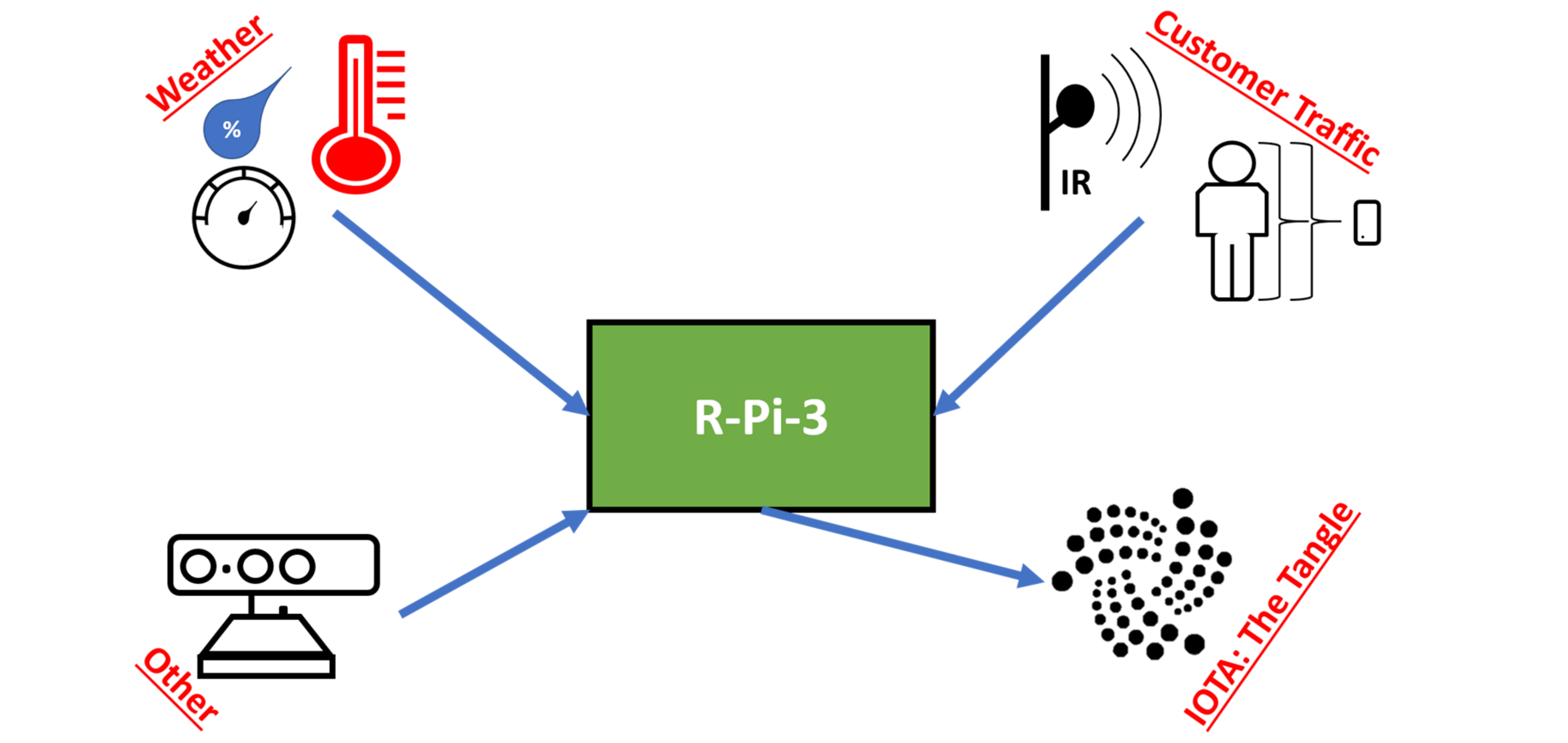

Figure 6. Diagram of the components involved in the environmental monitoring subsystem. The system acquires weather and customer traffic information and sends them to The Tangle. Future work will incorporate other sensors.

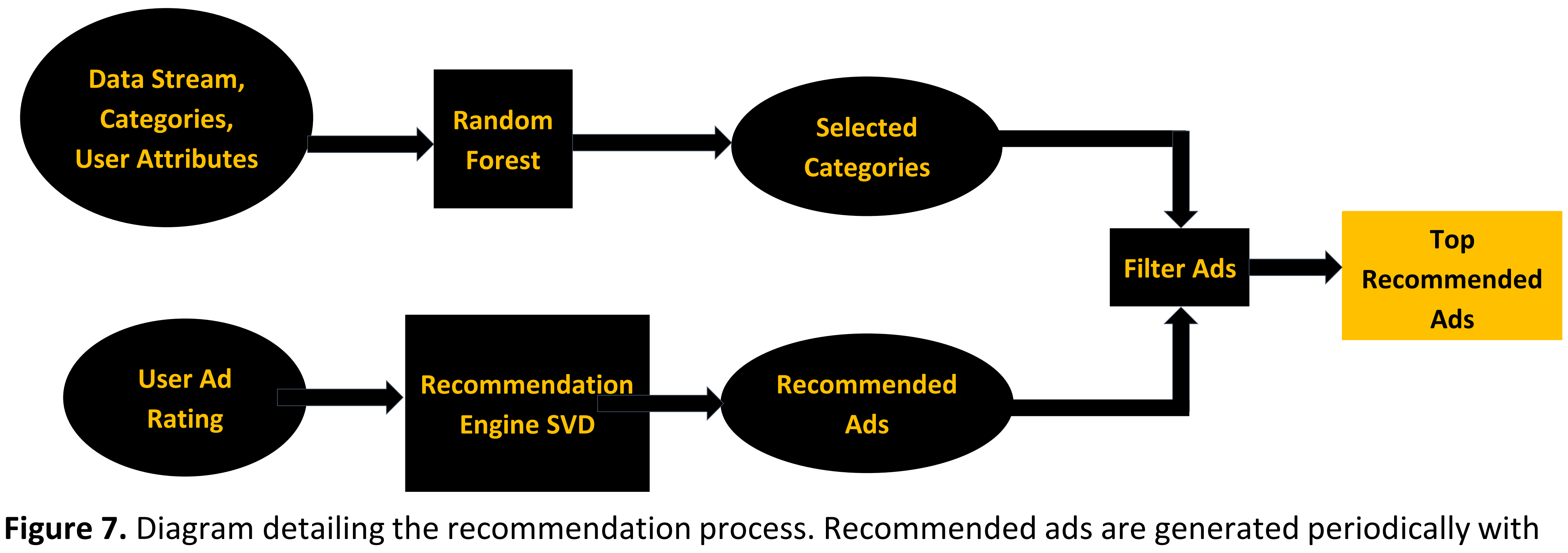

Figure 7. Diagram detailing the recommendation process. Recommended ads are generated periodically with the SVD algorithm popularized during the Netflix Prize competition. A random forest is trained with previous likes, user attributes, and data from environmental setting to predict relevant categories at ad retrieval time. The recommended ads are filtered to only show the most relevant categories.

Subsystem	Limitations	Future Work
Database	System heavy internet/network usage	Use an alternative database to Firebase.
Environmental Sensing	Limited number of sensors were used.	1. Incorporate more sensors. 2. Improve customer traffic monitoring.
IOTA Integration	The snapshot issues.	NA.
Kiosk	1. Kivy is limited to simple designs and renders slowly. 2. Raspberry Pi 3 couldn't handle computational demands.	1. Use a more powerful microprocessor. 2. Incorporate a key that users enter to log in.
Mobile App	1. Security: There is no user name or password, only facial recognition. 2. One can spoof the recognition.	1. Allow users to login again. 2. Add a password to the account with Facial Recognition as user name.
MS Cognitive Services	1. Large Person Group can only handle 1,000,000 people per group. 2. Requires pre-training a group each time a new user is added.	1. Use GPS location to filter out people in groups and create faster lookups. 2. Add person group training on a daily timer.
Recommendation System	1. Simulated and limited data. 2. Only 3 users. 3.. Running on 1 VM.	1. Collect more data. 2. Place recommendation system on a scalable VM network.

Table 2. Table of the critical limitations observed for each subsystem during the development of SenseAd and their potential future work. We would like to collect more data and improve the user interface of both the mobile app and kiosk.

Acknowledgements

We would like to acknowledge and thank the following: Professors Bai and Kuhl, Engineering Electronic Shop, Engineering Machine Shop, and the Electrical and Computer Engineering Department.

References

[1] Popov, Serguei. "The tangle (2017)."
[2] iota.org