Synthetix - System Architecture

General

Synthetix Mobile and Web applications architecture defines the behavior of the system and how its components interact with each other to achieve the desired functionality. The architecture of Synthetix plays a critical role in determining its success and popularity as it affects factors such as user's friendliness, functionality, scalability, maintainability, and performance.

Since the application main usage model is via a mobile application it's important to observe the best option to determine the technology. There are several architecture patterns that can be used to design Synthetix mobile application, including Model-View-Controller (MVC), Model-View-Presenter (MVP), and Model-View-View Model (MVVM). These patterns provide a blueprint for organizing the code and defining the responsibilities of each component.

The Model-View-Controller (MVC) architecture pattern separates the application into three components: the model, the view, and the controller. The model represents the data and logic of the app, the view displays the data, and the controller acts as a mediator between the two. This pattern is widely used and provides a clear separation of concerns, making it easy to maintain and test the app.

The Model-View-Presenter (MVP) architecture pattern is similar to MVC but has a slight variation. The presenter component acts as a bridge between the model and the view, handling the presentation logic and making it easier to test the app. MVP is a popular choice for mobile app development because it provides a cleaner separation of concerns, making it easier to maintain and scale the app.

The Model-View-ViewModel (MVVM) architecture pattern is a variation of MVP that is designed specifically for use with the Windows Presentation Foundation (WPF) platform. In MVVM, the view model component represents the state of the view and

provides data binding to the view. This makes it easier to update the view without affecting the underlying data model, making the app more flexible and maintainable.

In addition to these architecture patterns, there are several other factors that need to be considered when designing Synthetix mobile app architecture. One key factor is the app's performance, as slow-performing apps are often uninstalled by users. This can be achieved by using efficient database related algorithms and data structures, as well as optimizing the app for different device types and screen sizes.

Another important factor to consider is scalability. As the user base of an app grows, it is important that the app is able to handle increased traffic and data storage requirements. This can be achieved by using cloud-based storage solutions and employing a scalable architecture that can easily handle increased loads.

Synthetix web site architecture will be synchronized with its mobile application and so its design and organization, including the structure of its pages and the relationships between them. The web site map, which is a hierarchical visual representation of the system will be designed to ensure easy of use, high performance and user's friendliness.

Synthetix web site will be organized according to its content, determining the structure of individual pages, such as the use of headings, subheadings, and paragraphs, as well as the placement of images and multimedia. The content should be well-structured and easy to read, and should be optimized for search engines.

A key feature for Synthetix web site architecture is performance. This includes the loading speed of the site, as well as its responsiveness and ability to function correctly on different devices and screen sizes. To improve performance, we will use efficient algorithms, reduce the size of images and other media, and optimize the use of code and CSS. Another important aspect is accessibility ensuring that the site can be used by people with disabilities, such as those who are blind or have low vision. This can be achieved by using techniques such as providing alternative text for images, using descriptive links, and providing clear and concise navigation.

In conclusion, Synthetix mobile and web applications are designed to be integrated and synchronized together, offering same functionalities using both interfaces. Users

will be able to perform safe internet browsing, ensuring credentials and personal information privacy through mobile devices and internet browsers with ease of use and high performance.

Main components

1. User Input: This is where the user inputs the information they want to keep private, such as their real phone number and email address.
2. AI Model: This is the machine learning algorithm that generates the synthetic, fictitious identities. The AI model takes the user's input and creates a profile with a synthetic phone number, email address, and personal information.
3. Data Encryption: To ensure that the user's private information is secure, the AI model's output is encrypted before it is stored. This encryption helps to prevent unauthorized access to the user's data. All data transmitted between the user and Synthetix would be encrypted using a secure encryption algorithm, such as AES-256 or RSA. This would ensure that even if the data is intercepted, it would be unreadable without the encryption key.
4. Honey Encryption layer to protect user’s account security. This type of encryption enables a higher cybersecurity standards for password protection and data theft.
5. Database: The encrypted data is stored in a secure database. This database is accessible only to the Synthetix system and the user.
6. Web/Mobile Application: The user interacts with the Synthetix system through a web or mobile application. The user can view, manage, and update their fictitious identities. The application also ensures that the user's internet activity is anonymous and untraceable.
7. Network Communication: The web/mobile application communicates with the database and AI model over a secure network connection. This connection is encrypted to prevent unauthorized access to the user's data and activity.

8. Internet Anonymity: When the user is browsing the internet, their activity is anonymous and untraceable. The Synthetix system routes their internet traffic through the fictitious identity, protecting their real identity and personal information.

High Level Architecture

• Front-End: The user would access the application through a web or mobile interface. This would include a user-friendly dashboard to create, manage, and delete profiles.

• Back-End: This layer would be responsible for generating, storing, and maintaining the profile information. It would include APIs for accessing the data and processing requests from the front-end.

• Database: This layer would store all the profile information, including the phone numbers, email addresses, and other data.

• AI Engine: This layer would use AI algorithms to generate unique and realistic profiles.

Functional Design:

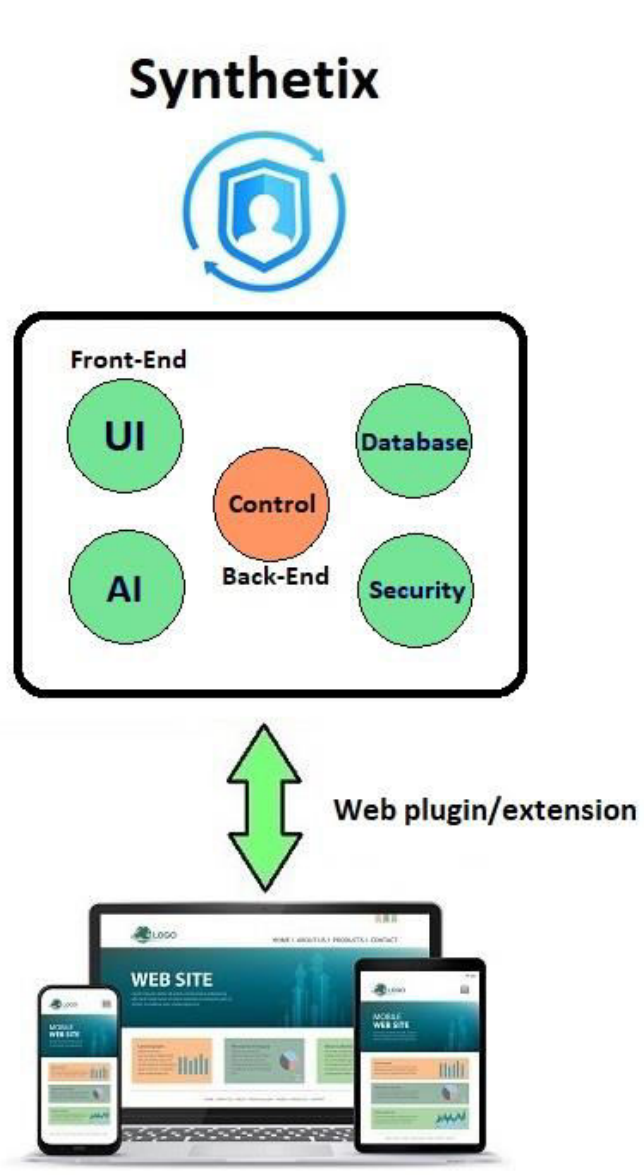
• Profile Generation: The user would be able to generate new profiles with unique phone numbers, email addresses, and other details.

• Profile Management: The user would be able to manage their created profiles and make changes to them.

• Profile Deletion: The user would be able to delete any profiles they no longer need.

• Data Security: The system would use encryption and other security measures to protect the user's data.

• AI-Based Profile Generation: The AI engine would create realistic profiles that are not easily identifiable.



System's High Level Architecture diagram

Tech Stack

Frontend programming languages will include:

- HTML, CSS, JavaScript for the user interface and design.
- React for the frontend framework.
- React Native or Flutter for the mobile app development.

Backend programming languages will include:

- NodeJS provides good performance, scalability, and the ability to handle large amounts of data. Express is a popular web framework for NodeJS and can handle multiple requests from users in real-time.
- A database such as MongoDB to store data.

The AI model will be developed using:

- The AI model for this system will be implemented using Python and the scikit-learn library. The Random Forest model is a good choice for this system, as it is efficient in handling large datasets and can be used to predict a user's activity based on their past behavior.
- Natural Language Processing (NLP) libraries such as spaCy or NLTK for text processing and analysis

Encryption will be achieved using:

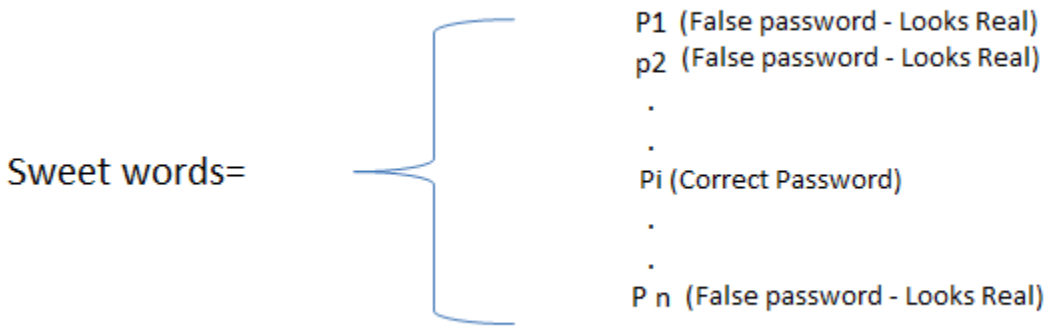
- SSL/TLS for secure communication between the client and server
- AES is a well-known and secure encryption standard that can be used to protect user information.
- The Honey Encryption Module can be implemented using the AES encryption algorithm, which would secure the key exchange while browsing the internet.

User interface will be designed using:

- Sketch or Figma for wire framing and prototyping.
- Adobe Photoshop or Illustrator for graphics and visual design.

Honey Encryption

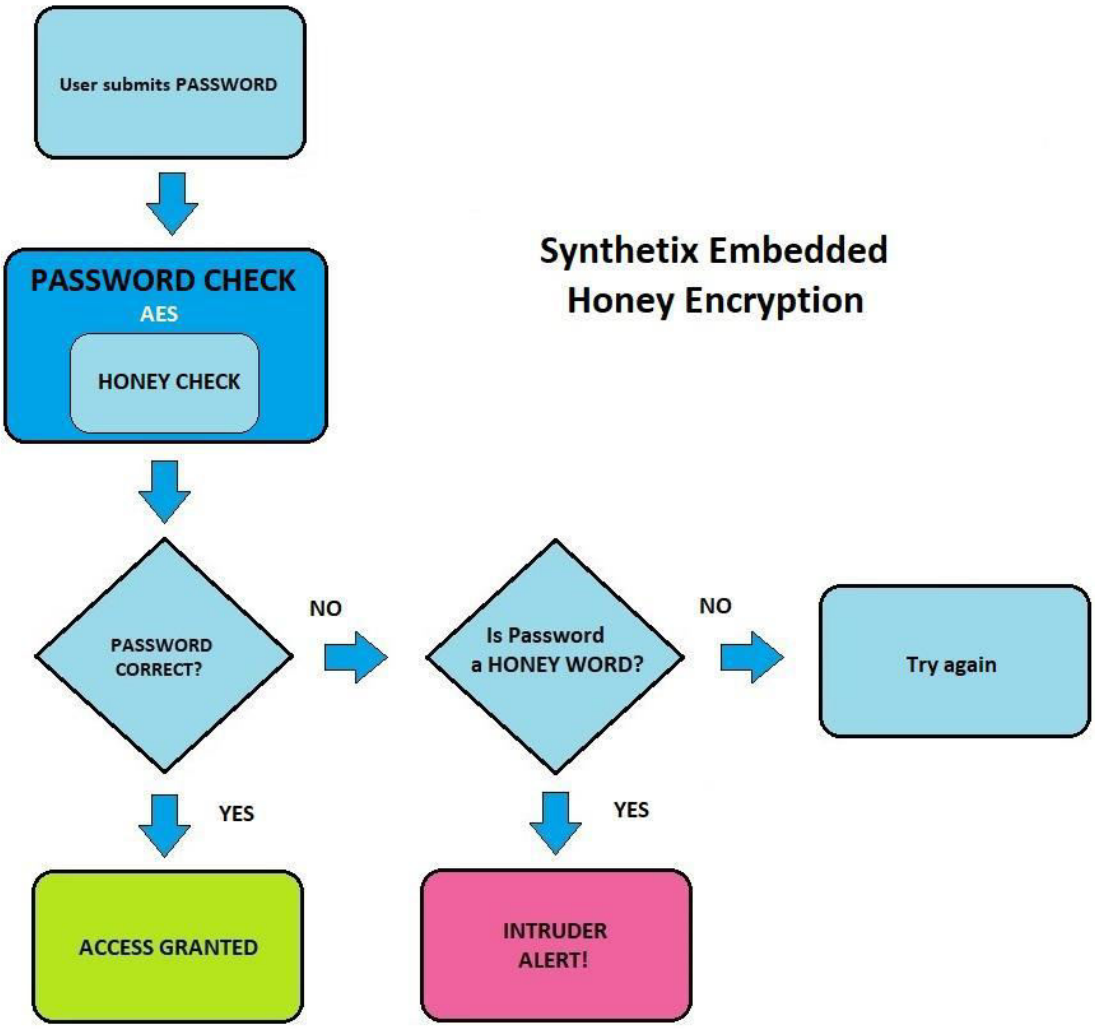
Honey encryption is a method used to distract and defend against brute force attackers. The scheme provides high resiliency level against attacks by ensuring that decrypted messages with invalid keys, yield a valid-looking message. Honey encryption provides a deceiving mechanism to increase the encryption strength and security measures. The concept of honey encryption is used to detect attackers when trying to decrypt data. The method introduces many passwords, and only one is the correct one, as all the rest are false passwords. These bogus passwords are called honey words or decoys. The entire list of honey words are called sweet words. False servers are set up to distract attackers and called honeypots. Honey words are false passwords stored in a hash table, that when stolen can detect and alert about an intrusion.



Typically, hackers are able to crack passwords by checking whether it is correct or not. But in case of honey encryption, when an intruder tries to brute force crack the encryption, he/she will be supplied with a password that imitates the real one. This confuses the attacker to differentiate the real password from the false one. Once the system detects a use of a honey word (False password), it will flag an intrusion. In

conclusion, using honey encryption on top of a conventional AES standards, enhancing the cybersecurity level for passwords authentication and data access.

The honey encryption system in Synthetix is adding an additional layer of security to the key exchange process when browsing the internet and accessing accounts. It works by using a technique called "decoys" or "honeypots" to lure potential attackers away from the real encryption keys. The algorithm creates multiple bogus encryption keys, which look like the real key to an attacker, but are actually decoys. The real encryption key is securely stored and only the user with the proper authentication can access it. In case of an attempted attack, the honey encryption system would detect it and direct the attacker to the false decoys, thus protecting the real encryption key and maintaining the confidentiality of the user's information.



Machine Learning Classifier

Synthetix is using Random Forest approach for its Machine Learning mechanism. The algorithm is trained using public datasets of fictitious names, email addresses and other personal information to build a synthetic user profile. Random Forest is a supervised machine learning algorithm that is used for classification and regression tasks. It is an ensemble model that combines multiple decision trees to make predictions. The algorithm is named "Random Forest" because it creates multiple decision trees using a random subset of the data and features, hence the term "Forest."

How it works? The decision trees in a Random Forest algorithm are constructed using a technique called bootstrapping, which involves randomly selecting observations from the training data to create new samples. Each tree in the forest uses a different sample of the data, and features are randomly selected from the feature set to split the nodes of the tree. This randomization leads to diversity in the trees created, and the diversity results in reduced overfitting and improved generalization performance.

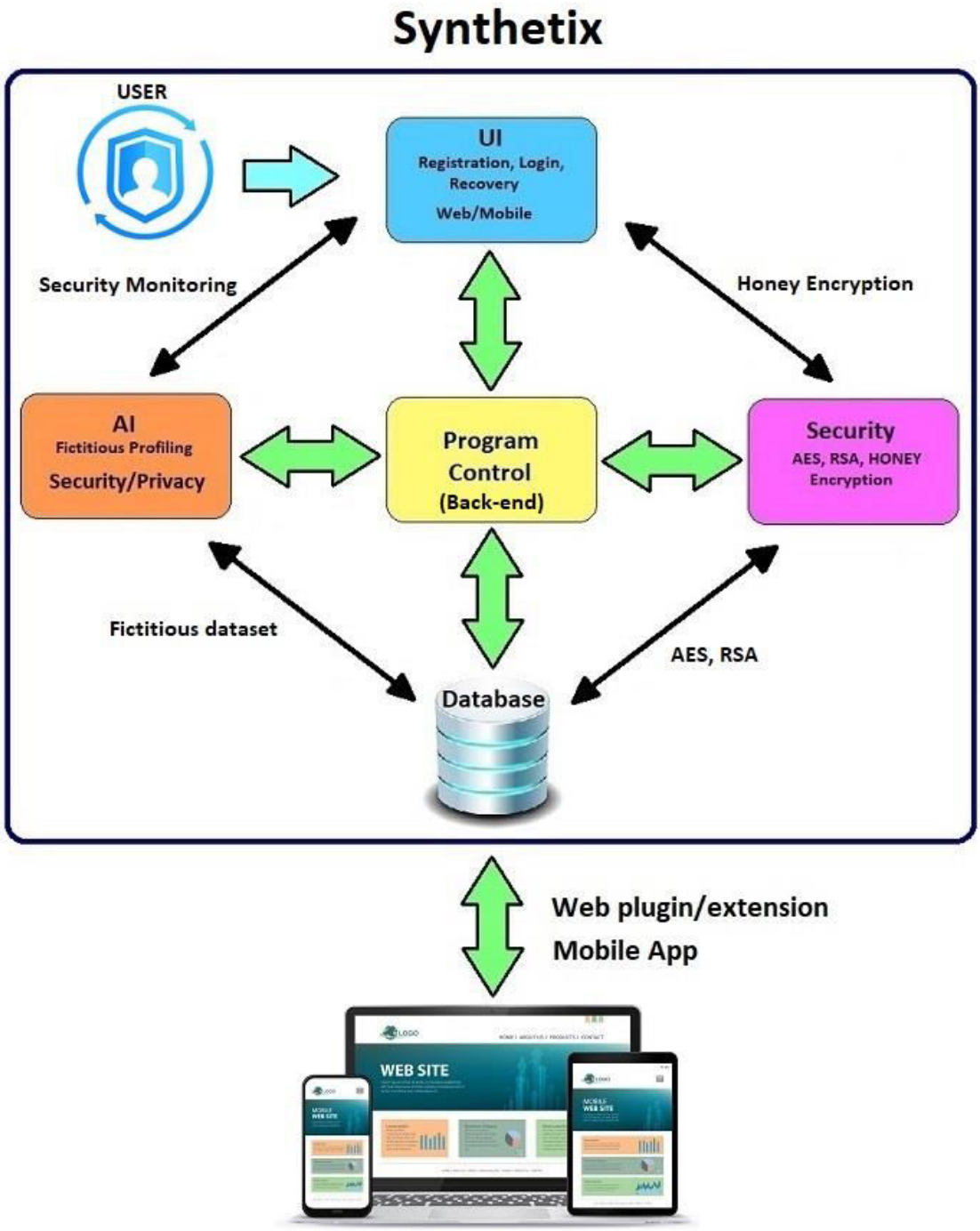
Once all the trees are constructed, the final prediction is made by combining the predictions from all trees in the forest. This is done by taking a majority vote in classification problems and averaging the predictions in regression problems.

Random Forest algorithms have several advantages over other machine learning algorithms like. The main advantages are large data handling and straight forward implementation. Another important advantage of Random Forest algorithms is their ability to handle non-linear relationships between the features and the target variable. This is achieved by the combination of multiple decision trees, each of which can model a different non-linear relationship.

Their main limitation is the fact that they can be computationally expensive, especially when there are large numbers of trees in the forest. This may cause performance issues but in Synthetix application the amount of data points is not large enough to create a major issue.

Synthetix Block Diagram

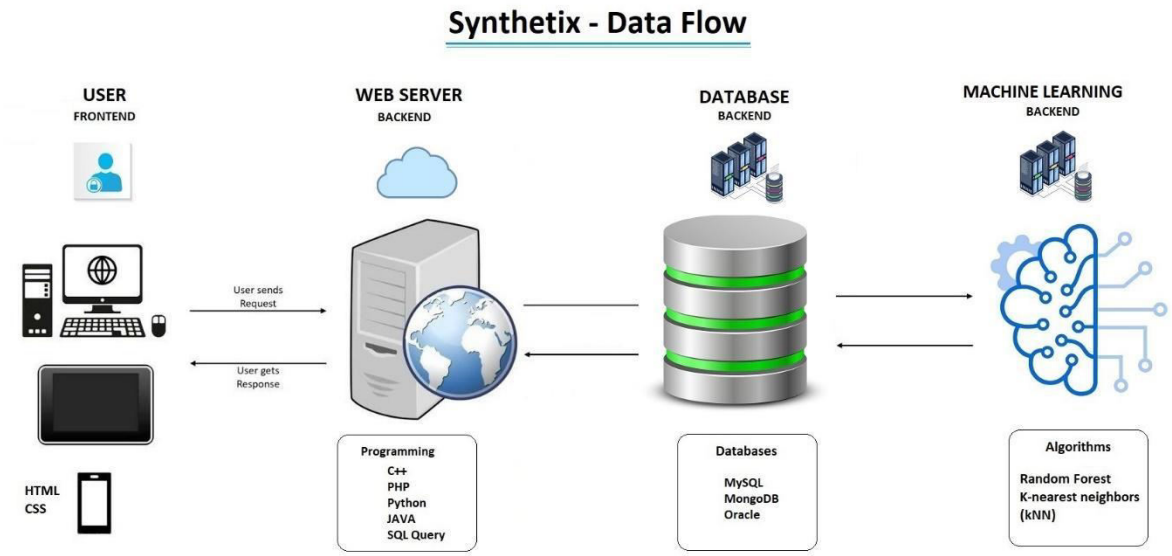
Synthetix block diagram is a visual representation of the application, showing the flow of data and control structures. It is a graphical representation of Synthetix logic, operations, functions and data flow. Each block represents a component/module and its interaction with the other parts of the application. The diagram presents the application's structures, the flow of data, and the order in which operations are executed. It also presents Synthetix interaction with web/mobile applications.



Synthetix block diagram

Synthetix Data Flow

Synthetix data flow diagram describes the movement of data through the different stages of the application. It represents the process of data transformation through a series of steps and operations. The data flow diagram is visualized as a series of connected components/nodes, each representing a step in the data processing pipeline.



Synthetix Machine Learning Flow

Synthetix machine learning flow represents the sequence of steps involved in creating and implementing its machine learning model. It describes the process of building, training, and using the model to make predictions based on input data.

The machine learning flow includes the following:

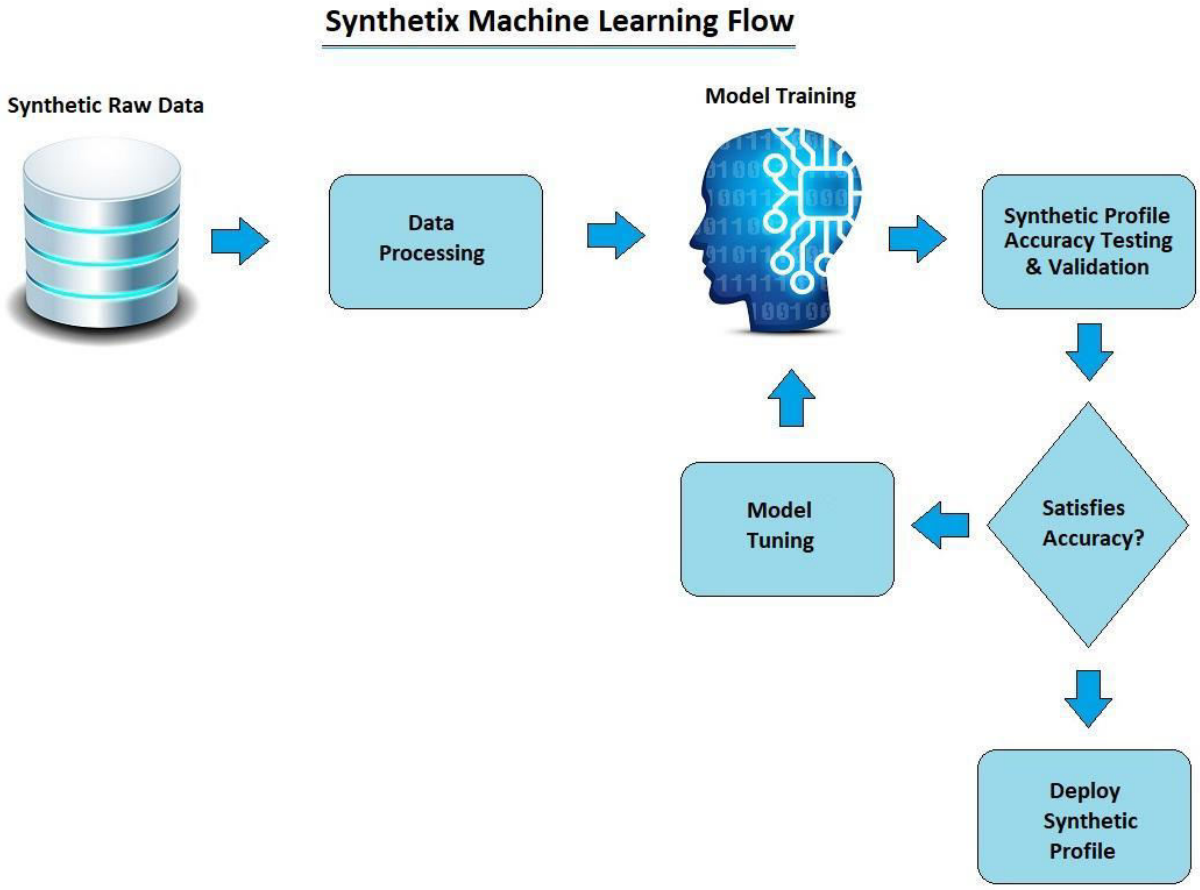
Raw Data collection: This is the data that will be used to train the machine learning model. This data can come from various sources, such as databases, files, or web APIs.

Data processing: In this stage, the collected data is cleaned, transformed, and pre-processed to make it suitable for training the model. This can include tasks such as handling missing values, scaling the data, and splitting the data into training and test sets.

Model training: The selected machine learning algorithm, Random Forest, is then trained on the prepared data. The algorithm updates its parameters to optimize its performance on the training data.

Model evaluation (Testing and Validation): The trained model is evaluated on a test set to measure its accuracy and performance. This stage provides insight into how well the model generalizes to new data. We use Model Tuning to provide accurate results.

Model deployment (Synthetic Profile): If the model performs well in the evaluation stage, it is deployed into Synthetix production environment and used as a user's synthetic profile.



Conclusion

Synthetix technology architecture includes advanced modules of UI, database, ML and cybersecurity. These modules are integrated using web and mobile application multi-disciplinary development environment. The architecture is designed to provide a high performance, efficient and highly secured system to synthesize user's profile, to protect private and personal information. It takes into consideration accountability, data security and liabilities using the most up-to-date, state of the art computer science methods and approaches. The end result is a comprehensive privacy protection and safe internet browsing system, ensuring secured online

activities, preventing data theft prevention, and eliminates unwanted advertisements and spam.