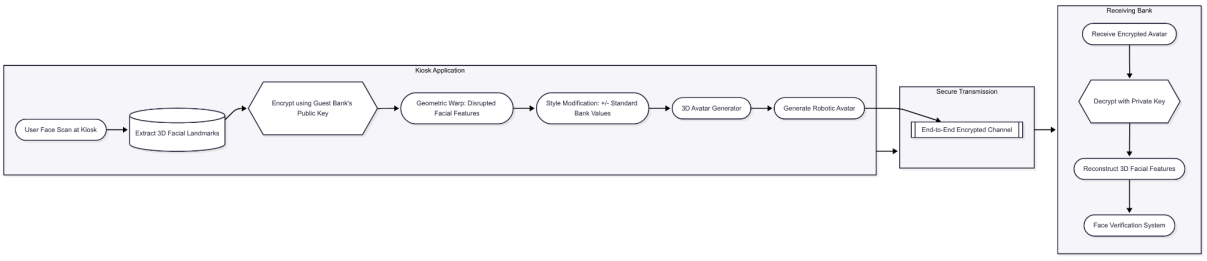
1. The payee mobile's bank application is being opened and authenticated by them. Then they can find a option called "GMP" (Guest Mode Payment). Payee needs to click that option to open the Kiosk mode.

   Kiosk mode is a configuration that locks a device to run only a specific application or limited set of functions, commonly used in public or controlled environments like ATMs, information kiosks, or exam systems. It preserves user privacy by restricting access to system settings, apps, and local storage, ensuring users cannot alter configurations or retrieve sensitive data. Sessions are typically isolated and automatically reset after use or inactivity, clearing any user input, browsing history, or temporary files.

   The payer can find the details like Payee details and some fields to enter the amount, choosing the bank accounts, face scanners.

2. In the kiosk application, the user's face is initially scanned and processed to extract 3D facial features. These features then undergo a geometric transformation, where the extracted features are encrypted using the guest bank's public key – this public key must be shared in advance as part of the agreement between the kiosk operator and the target bank. This encryption process results in distorted, unrecognizable facial features. Next, a style transformation step is applied, where the encrypted features are further transformed by adding or subtracting standardized values defined by the bank, improving privacy and ensuring consistency in the data format. These transformed features are then sent to a 3D avatar generator, which creates an abstract avatar based on the transformed data – such as a robot representation. Once created, a secure end-to-end encrypted communication channel is established between the kiosk and the bank, through which the avatar is sent in an encrypted format with the bank's public key.



3. Once the data is received, the bank can reverse the changes – first decrypting the avatar to restore the modified features, undoing the style changes, and finally decrypting the geometry data using its private key to restore the original facial identities. This allows the bank to accurately verify the user's identity while preserving privacy throughout the transmission and processing stages.
   Then the Source-Popup Window will be opened once it is successfully authenticated. Meantime the bank verifies the funds available in tokens and select a token to push the payment authentication, along with 1 fake token for security purpose.

4. Now the guest user needs to select his token in the first attempt else the option will be locked. Then the guest enters the token key for the authentication and the details are sent to the guest bank’s server for validating the transaction.

When verifying a user's bank PIN or facial recognition data, the client must send a secure payload to the bank's backend over HTTPS. For PIN verification, the payload typically includes the user ID, the authentication type ("pin"), a hashed version of the PIN (e.g., using SHA-256 or Argon2), an optional salt if client-side hashing is used, and a timestamp to prevent replay attacks. For facial recognition, the preferred method is to send a face embed (a numeric vector generated using models such as FaceNet or ArcFace, typically 128 or 512 dimensions) with the model version and timestamp. Alternatively, in less secure or fallback situations, a base64-encoded image can be sent. For facial recognition, the payload typically includes the user ID, the authentication type ("face"), and the embed or image. On the bank's side, the server securely compares the hashed PIN with a stored hash, or matches the received facial embedding with the stored embeddings using cosine similarity or Euclidean distance. All data must be transmitted securely, and authentication payloads must be designed to support extension and protect against replay attacks through timestamps or non-timestamps.

5. Once the user is verified, the originating (payer’s) bank assembles the payment request with essential details such as sender and recipient account numbers, routing numbers (ABA), amount, purpose, and a unique transaction reference. For ACH transfers, the bank sends the request in batch to the **ACH network** (either FedACH or The Clearing House), where it’s cleared and settled typically within 1–2 business days. For faster settlement, banks may use **Fedwire**, which allows real-time gross settlement for high-value or time-sensitive payments, or the **RTP network** for instant 24/7 payments. The payload is formatted according to standards like **NACHA (for ACH)** or **ISO 20022 (for RTP/FedNow)**, often with additional security measures like digital signatures or tokens. Once processed, the receiving (payee’s) bank credits the funds to the recipient's account and sends an acknowledgment back to the originating bank, completing the transaction. Confirmation is then displayed to the user.