# WIZZIT AUTHENTICATOR

AUTHENTICATOR WHITE PAPER

## The Background to the WIZZIT Authenticator

### THE EVOLUTION OF AUTHENTICATION

At its most basic level, bank grade authentication is built around a simple concept of the person being authenticated having something unique that is known to the authenticator (e.g. a credit card) which is then combined with something only known to the person being authenticated (e.g. A PIN) in a secure, encrypted and known format (e.g. via a POS device).

In the world of payments this was not complex where a POS device exists in the physical world, but got really complicated in the e & m-commerce world when the card was not present and a PIN could not be securely entered. Various methodologies emerged to counter this such as 3DS. However, this introduced new complexities such as managing the performance of an out of band SMS - and besides it came with the weakness that the OTP could be intercepted when MNO/Telco networks were involved through a simple SIM swop.

Security methodologies evolved to allow the push of the OTP via more secure methodologies such as Push USSD but the issue is that the message was simply being pushed to a pre-registered device - ownership of which could not be proven and as such the key principle of involving “something I have with something only I know” was fundamentally broken.

Again new technologies emerged to address this– a solution was developed that used a secured MNO’s Wireless Internet gateway to talk directly to the keys of the SIM, allowing the customer to enter their actual ATM/POS PIN directly into the handset. This overcame the aspect of PIN entry and both Visa and MasterCard along with the local payment authority (PASA) approved a new transaction type – called AMT – which saw non-repudiation in a similar manner to that of a physical POS. Further iterations evolved with similar security methodologies for use in the App world and this has so far proved successful within MasterPass.

Parallel to these developments the world of mobile channels has also evolved. Customer usage of Apps has not been the much-touted success – with customers using a very limited subset of their downloaded Apps. In addition, the move towards Instant messaging or Chat as it is known has come to dominate messaging – with dramatic declines in SMS and voice usage being noted.

The dependence on WIFI has seen many mobile phone users not even making use of an MNO/Telco network anymore – therefore not making use of SMS or USSD. With Chat maturing and able to provide a more relational interface without the requirement for a customer to download an App the next challenge of authentication arises.

## THE INTRODUCTION OF CHAT

Most Chat applications provide some form of encryption of their messages, but this is not typically to the level that trusted entities such as banks, cards, governments and others will accept. Not as a result of the absolute technical ability of this encryption but because the encryption processes places control of credentials and keys outside of these entities.

Banks for example, want to be able to be the final arbitrator of the authenticity of their own customers, using their own key structures and are not willing to pass this control on to the Chat providers. When it comes to matters of money - authentication is an absolute!

## AUTHENTICATION IN THE CHAT CHANNEL

The WIZZIT Authenticator provides a bi-directional bank grade encryption process for the most popular Chat platforms such as WhatsApp, Facebook Messenger, Viber, Telegram and others giving trusted entities such as banks the ability to securely authenticate their customers using their own credentials.

The two-way process allows encrypted data to now be sent to the customer as well to receive it – providing new methods of data sharing and authentication – whilst also making use of existing ones without the current worries and concerns. Working both on an MNO’s/Telco’s network as well as Wi-Fi - the days of Sim swop and the associated fraud are a thing of the past.

With a highly flexible customer interface the WIZZIT Authenticator is designed to allow businesses to deploy a secure authentication interface quickly and very cost effectively. The WIZZIT Authenticator can be provided as a turnkey service offering – hosted and operated by WIZZIT or it can be hosted by the customer. WIZZIT can supply all the hardware required.

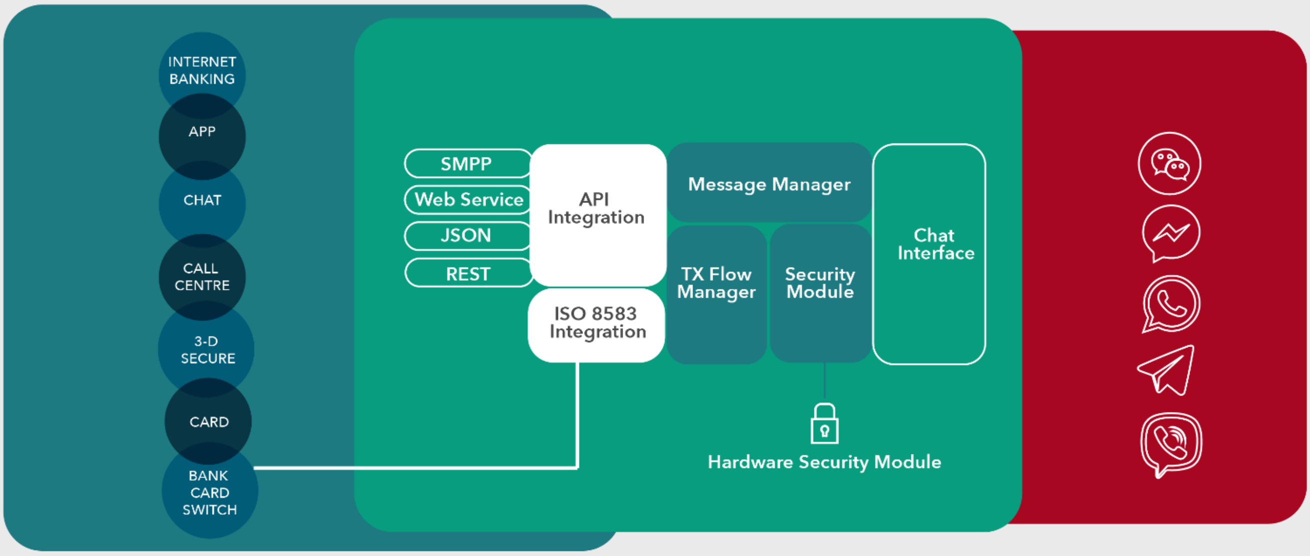
WIZZIT Authenticator has been subjected to rigorous international penetration testing and provides a strong base for compliance requirements such as PCI-DSS. At its core is a bank grade authentication methodology allowing rapid payments, banking and other security applications to be deployed within the Chat world.

## OTHER APPLICATIONS

Imagine a world in the future where one can securely register for and vote using Chat - order my prescription medicine – pay using my bank PIN – have access to my bank data via a third party – get my bank updates in Chat – approve a debit order – manage my home automation – KYC  myself – get my OTP (for anything) – get my PIN sent to me – receive my bills securely – and pay for them - imagine!

## ARCHITECTURE

The Architecture is designed as a stand-alone application that can be accessed by the development team via a set of API interfaces. An example of this shown below would be the integration of the solution into a banking environment.



**The API Integration that can be used to access the Authentication services is as follows:**

- SMPP to replace the SMS gateway and send request and authorisations via Chat
- Web service, JSON and REST API for the integration of the Chat application and other services that may require authentication.

**The system has an ISO 8583 Host to host node that allows for the connection of the authenticator to the banks card system to do the following for example:**

- Debit and Credit card PIN authorisation
- PIN Selection by the customer on the issue of a new card
- Changing of the card PIN.

The message manager takes the input from the API and manages the format of the message and OTP or if required directs this to the ISO 8583 Interface.

The transaction flow manager, as the name suggests manages the sequence and the response based on the specific transaction type.

The security module interfaces to the Hardware security module and ensures that the encryption of the sensitive data is handled correctly.

The Bank will load their security keys on the HSM to ensure that all secure transactions are end to end encrypted.

The Chat Integration allows the system to push and receive messages from the various chat platforms. It also manages the interaction of the secure PIN pad for the capturing of sensitive data.

**The following Chat interfaces are supported:**

- WhatsApp
- Facebook Messenger
- Telegram
- Viber
- WeChat

## INFRASTRUCTURE

Two separate Data centres are required to host the application. Both data centres need to be PCI DSS Compliant. Each data centre will host the application in its own secure VLan behind the banks firewall.

**The following is required for each data centre:**

- Safenet Protectserve HSM
- x86 4 Core server with 64 gig ram and 500 Gig storage

## TRANSACTION SCENARIO

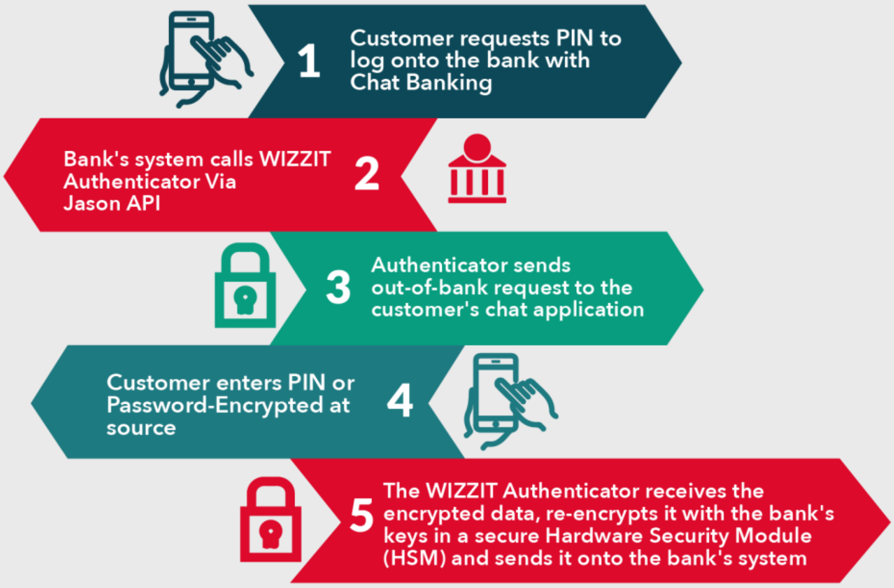
The following transaction flows are indicative of what can be done using the authentication channel.

### Scenario 1: Customer Logon

This method is useful for

- Chat logon
- Internet banking logon

**The Process below shows the interaction between the Customer, the Bank and the WIZZIT Authenticator when a customer wants to login to their Chat banking or internet banking platform**


1 Customer requests PIN to log onto the bank with Chat Banking
2 Bank's system calls WIZZIT Authenticator Via Jason API
3 Authenticator sends out-of-bank request to the customer's chat application
4 Customer enters PIN or Password-Encrypted at source
5 The WIZZIT Authenticator receives the encrypted data, re-encrypts it with the bank's keys in a secure Hardware Security Module (HSM) and sends it onto the bank's system

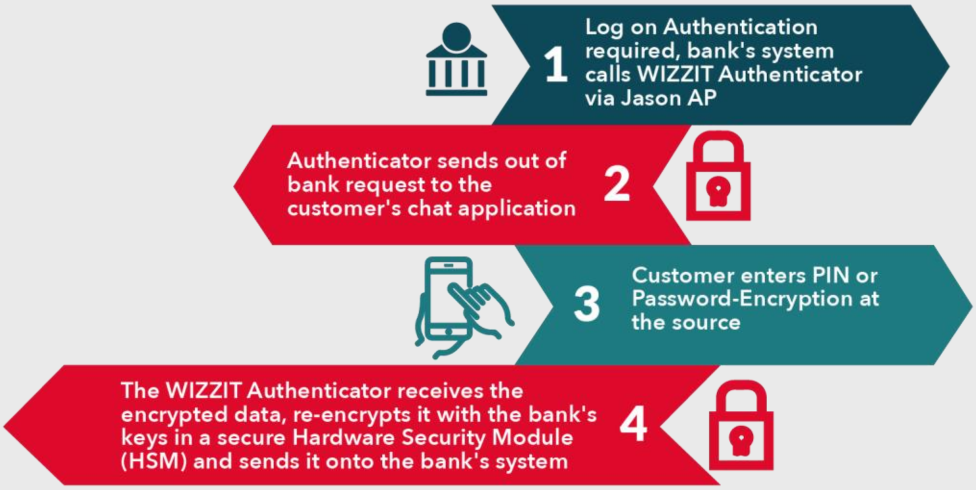
**A customer requests a PIN to login to their chat application:**

1. The Banks chat application sends a request for the PIN to the Chat application server.
2. The bank responds from the chat server via the API to the WIZZIT Authenticator.
3. The Authenticator Pushes an out of band push request containing a keypad. This allows the transaction to be encrypted at source.
4. The customer enters the PIN / Password which is encrypted using a DUKPT (Derived Unique Key Per Transaction) Method as well as the Authentication of the device, to counter the Man in the middle attacks.
5. The Authenticator receives the encrypted information. This information is sent to the HSM (Hardware Security Module) and encrypted with the Banks keys. The re-encrypted information is then sent to the bank. The bank at this stage can now decrypt or validate this information to allow the logon to proceed.

**Scenario 2: System pushes to customer for authentication on chat**

The scenario is the same as the one above except that the transaction is not initiated by the user. The initiation of the transaction is from the system. This method is useful for:

- Alerts for verification where a transaction is done over a certain limit.
- Alerts raised by the Fraud engine and, an authentication request is required, an authentication request can be pushed automatically to the customer.
- Where a call centre agent needs to verify the identity of a customer.
- Where a password needs to be changed by the user.


1
Log on Authentication required, bank's system calls WIZZIT Authenticator via Jason AP
2
Authenticator sends out of bank request to the customer's chat application
3
Customer enters PIN or Password-Encryption at the source
4
The WIZZIT Authenticator receives the encrypted data, re-encrypts it with the bank's keys in a secure Hardware Security Module (HSM) and sends it onto the bank's system

1. The organisations application that requires authentication will, at that point call the WIZZIT Authenticator via the API to trigger the request for Authentication to the client.
2. The Authenticator PINpad is then sent to the customer handset and displayed on their handset.
3. The customer enters the PIN or Password required in the PINpad. The information is then encrypted using the Organisations keys in the HSM.
4. The Authenticator receives the encrypted data and sends it on the bank or organisation for validation.

**Scenario 3: Utilising the application for card authentication with integration directly to the Bank's card switch**

This process is useful wherever the card PIN is required to authenticate a client, or a card PIN needs to be changed or issues. The push transaction can be system or call centre initiated. The same process is followed for Push transactions mentioned above but has the ability to do authentication of credentials to the bank's card switch.

The API allows for the bank to provide the Customer's mobile phone details as well as the card that needs the number validated.

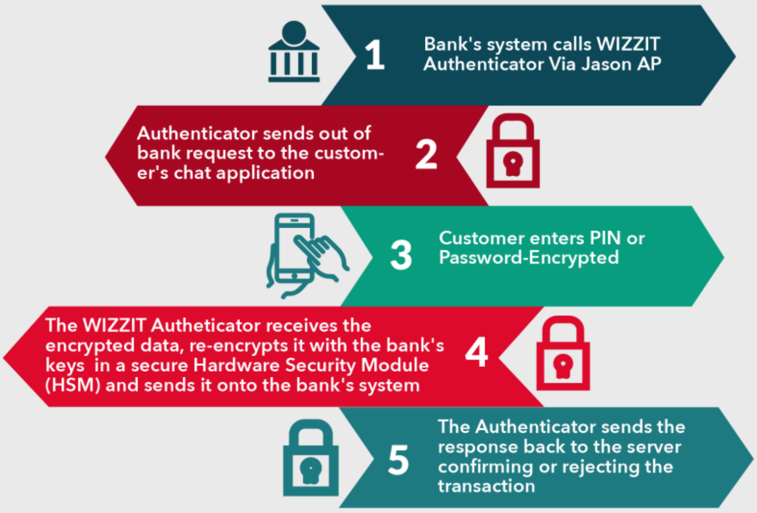
The Bank will have loaded their session keys on the WIZZIT HSM as well as the PIN block utilised. Once the customer is requested to authenticate a transaction or authorise using their PIN, The WIZZIT Authenticator will process the transaction directly to the bank's card switch allowing for the authentication of the customer's PIN.

The benefits of this are that the customers already have a pin number that they already know which will enable identification and authorisation.

**Examples of where this can be used are as follows:**

- Enable the customer to change customer PIN number via chat.
- To push a PIN pad to the customer to select their PIN on the first issue of a card.
- To use the PIN to validate card transactions done via a card.
- As a step-up authentication for tap and go transactions.
- Authorisation of Masterpass and mVisa

**As an authentication method in addition to the ACS for 3-D Secure**



1 Bank's system calls WIZZIT Authenticator Via Jason AP

2 Authenticator sends out of bank request to the customer's chat application

3 Customer enters PIN or Password-Encrypted

4 The WIZZIT Autheticator receives the encrypted data, re-encrypts it with the bank's keys in a secure Hardware Security Module (HSM) and sends it onto the bank's system

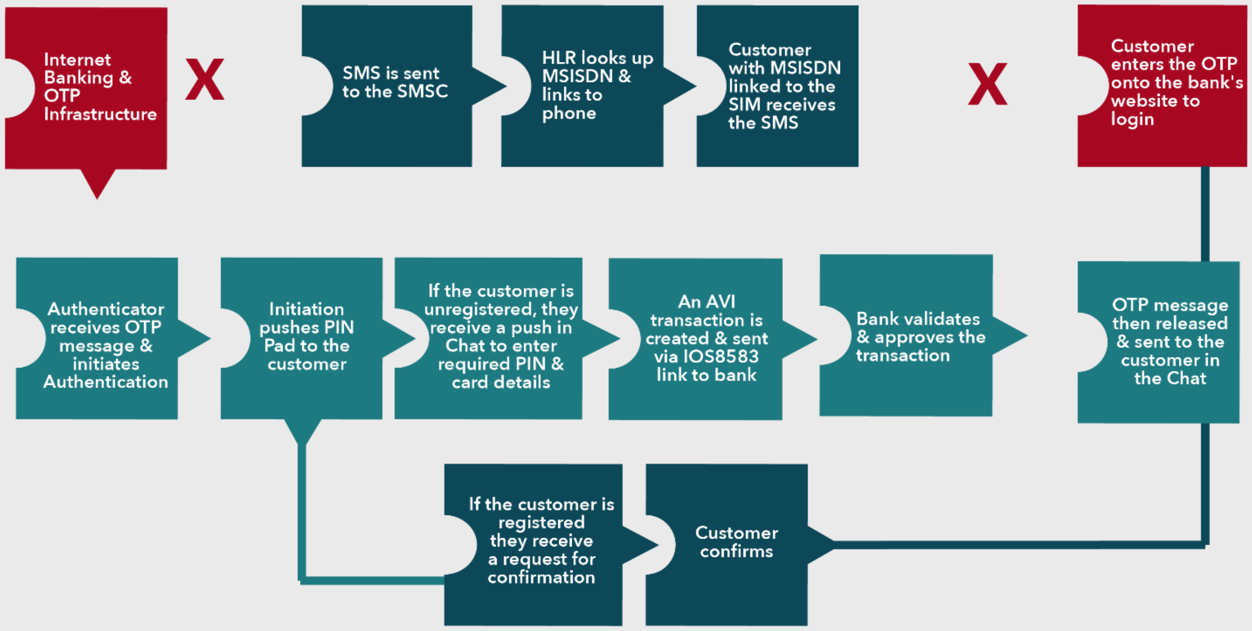
5 The Authenticator sends the response back to the server confirming or rejecting the transaction

1. The API is called by the Banks system when the Authentication is required.
2. The Authenticator pushes the message to the customer containing the PIN pad
3. The Customer opens the PIN pad and enters the Pin which is then encrypted under the banks keys.
4. The Authenticator receives the encrypted PIN. Creates an ISO 8583 AVI transaction and sends this to the Banks Switch

**Scenario 4: Internet banking login – eliminating issues around Sim Swop**

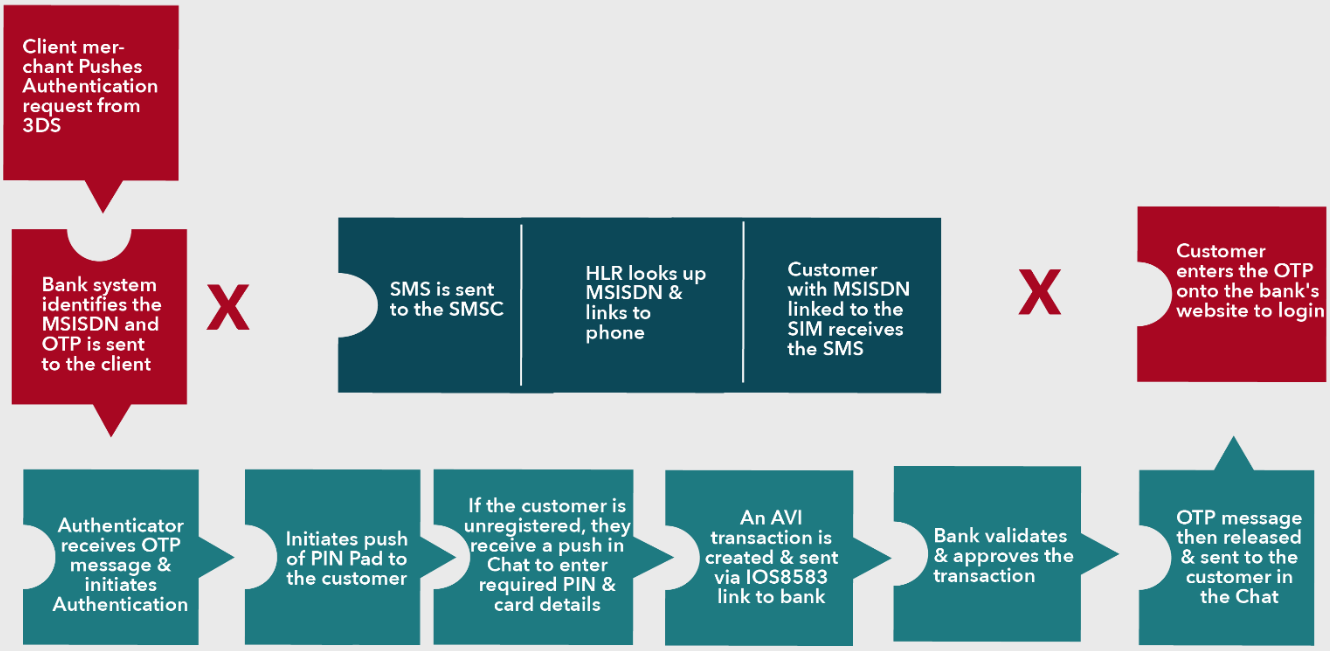
Logon will allow a developer to call the API to present the PIN pad. The SMPP 3.3, 3.4 and 5.0 are supported in order for the institution to seamlessly replace their current SMS gateway integration with a secure chat based encryption capability without needing to change any of the current OTP infrastructure that is in existence and reduce the integration timeframe for the bank significantly.

An example of this integration would be the integration of the solution in the internet banking and transaction environment of a bank to eliminate the risk of SIM Swap fraud.


Internet Banking & OTP Infrastructure
X
SMS is sent to the SMSC
HLR looks up MSISDN & links to phone
Customer with MSISDN linked to the SIM receives the SMS
X
Customer enters the OTP onto the bank's website to login
Authenticator receives OTP message & initiates Authentication
Initiation pushes PIN Pad to the customer
If the customer is unregistered, they receive a push in Chat to enter required PIN & card details
An AVI transaction is created & sent via IOS8583 link to bank
Bank validates & approves the transaction
OTP message then released & sent to the customer in the Chat
If the customer is registered they receive a request for confirmation
Customer confirms

### Scenario 5: Account Verification Instruction

An additional application could also be used for the validation of Authenticated Credit Transactions via and AVI (Account verification Instruction) transaction. The same process can be used for any validation that is done for 3D secure to directly replace the OTP or PIN validation.


Client merchant Pushes Authentication request from 3DS
Bank system identifies the MSISDN and OTP is sent to the client
X
SMS is sent to the SMSC
HLR looks up MSISDN & links to phone
Customer with MSISDN linked to the SIM receives the SMS
X
Customer enters the OTP onto the bank's website to login
Authenticator receives OTP message & initiates Authentication
Initiates push of PIN Pad to the customer
If the customer is unregistered, they receive a push in Chat to enter required PIN & card details
An AVI transaction is created & sent via IOS8583 link to bank
Bank validates & approves the transaction
OTP message then released & sent to the customer in the Chat

For the ACH transactions a process authentication can also be used for this purpose. If an OTP or in app validation is used this can be used as a direct integration to the authenticator to replace this process with a more secure option.

## CONCLUSION

For further information or to arrange a demonstration and discussion, please call us Contact us.

### WIZZIT INTERNATIONAL

- Dirkb@wizzit-int.com
- Davep@wizzit-int.com
- Charlesr@wizzit-int.com
- Brianr@wizzit-int.com

**Europe**

- Gideonv@wizzit-int.com

**Australasia**

- Nicholasr@wizzit-int.com

**London**

- Simonellis89@icloud.com