Features of Personal SDK and Personal Mobile

This article describes key features of the ready-to-use mobile app Nexus Personal Mobile and the software development kit Nexus Personal SDK, which can be used to implement your own mobile app for authentication and signing. Personal Mobile is entirely based on Personal SDK.

Both the SDK and the app comes with a complete protocol and interface documentation.

Features

Features	**Pe S**
Use cases	
Activation of user profiles, including provisioning of user certificates for authentication, signing, and encryption.	
Authentication to local or web applications.	
Signing transactions.	
Certificate import and renewal.	
Delete profiles from device, both started from server and local	
SDK app branding	
Public keys, certificates and other identity metadata are available to the app.	
Implementer decides which identity and other parameters shall accept or reject the pending request.	
Implementer-specific metadata can accompany any request, for example raw data, text, pdf or images.	
Attestation key can be provided by implementer so that the server can validate that it is your client responding.	
Built-in fingerprint and biometric authentication.	
Easy-to-use and intuitive interface.	

Registering device and receiving push notifications from Nexus Push Service hosted by Nexus.	
Hosting your own Nexus Push Service backend server for push notifications.	
Displaying the SDK licence dependencies.	
Handling app-to-app transitions using the 'personal://' URL-scheme for external applications.	
Built-in mobile device management (MDM) integration. This applies to iOS only.	
Secure sharing of keys with apps signed by same developer via shared key chain. This applies to iOS only.	
Secure communication	
Activation links are only for one-time use, and cannot be reused.	
PIN codes are validated on the server side, to perform flow control and add extra security.	
The identities continue to communicate with the same server that provisioned them.	
Prevention of man-in-the-middle attacks by TLS handshake and server certificate validation in response.	
Possibility to define that specific server certificates are the only ones allowed.	
Attestation key included to make sure that the client is genuinely Nexus.	
Secure key storage	
Generates keys on the device and provides proof of possession to the server.	
Key storage is device-bound and non-extractable.	
Protected with obfuscation, root detection, real-time checks and debugger detection.	
When storing secrets offline, device keystore is a required part. Android 6+ is required for OTP.	

Minimum PIN policy is fixed at six digits and disallowing sequences.	
Blocked after wrong PIN attempts for increasing amount of time, until the tenth try when the identity is entirely blocked.	
Lifecycle management	
Uses either X.509 certificates or raw key pairs, based on JSON Web Keys, see RFC 7517. When activating a certificate, a signed PKCS#10 certificate signing request (CSR) is provided for each key in the activation response.	
Renewal of certificates supported, including cryptographic key exchange.	
Secure import of keys is supported: • Import keys from the server side, for example for encryption certificates. • Import keys to the keystore of the device's operating system.	
Identities can be migrated from one server to another, but keys never leave the device.	
Support for securing OATH tokens for use in offline scenarios, for example with bad internet connection, RADIUS or on airplanes.	
Usability	
Uses either Nexus Hybrid Access Gateway, Nexus PRIME or Hermod to communicate.	
One server implementation can talk to all our clients: iOS, Android, Windows, Mac, and Linux.	
Possibility to have multiple identities in the SDK simultaneously.	
Support for multiple simultaneous authentication or signing requests.	
Possibility via server trust to login to external servers by trusting the certificate authority (CA).	
Uses standard protocols like HTTPS, JOSE and REST. All keys and crypto are handled within JOSE standard objects.	
Support for Google OTPAUTH protocol. This enables migration from Google and Microsoft Authenticator. Support for user display name in OTP profiles for ease-of-use.	
Possibility to secure your existing accounts with two-factor authentication, for example in Google, Visma, Hubspot and Microsoft.	
Cryptographics	

Minimum 2048-bit RSA key pairs.	
Signatures use standard JSON Web Algorithms (JWA), either RS256 or RS512. For more information, see RFC 7518.	
Keys are stored with password-based key derivation and encrypted using Advanced Encryption Standard (AES). Keys use device keystore when available.	
Keys are securely encrypted with multiple layers of AES-256.	
Keys are stored with server-based parameters to increase security in online scenarios.	