okay

Is your app ready for PSD2 SCA compliance?

Tick this simple checklist to assess your app compliance with Strong Customer Authentication.

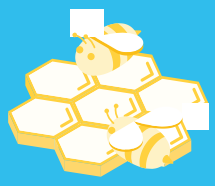
Are you certain that..	Why it is important	How we can help
☐ Transactions and authentications happen in a **separate secure execution** environment.	§9 of RTS requires "the use of separated secure execution environments through the software installed inside the multi-purpose device;"	On Android we provide security through having unique one-time use codeblocks delivered just-in-time to the device, with transaction details and all required code. iOS is of course also supported.
☐ The **posession** / ownership of the device is verified for each use.	§7 of the PSD2 requires that PSPs mitigate "replication of the possession factor". This implies that the integrity of the device must be verified.	We check multiple values in order to verify that the execution environment has not changed. The checks are directly linked to the secure storage, so that if an attacker manages to access the data storage they'll not be able to decode it.
☐ The **authenticity, integrity and confidentially** of everything displayed to the user is verified.	§5 of the PSD2 requires this for all authentication phases, including the display of transaction information.	Our trademarked «what you sign is what you get» mechanism builds user interfaces, then verifies them throughout the authentication process.
☐ There is a dynamic link between payment details and user identity which is kept throughout the transaction.	This dynamic link is required in §5 of the PSD2.	With Okay transaction details are transferred in obfuscated code, displayed as an invisible watermark on the user interface, and analysed server-side.
☐ SMS is **not** used for one-time-pin.	Using SMS is not strong enough to prove possession, as it is not communicated securely, or protected from malware, as required by PSD2 §6-8 and RTS §4-5.	With Okay we provide an SDK that provides much stronger security than you get with SMS.
☐ All transaction related interactions with users are **tracked and logged.**	§72 and §73 of the PSD2 and §29 of the RTS require the PSP to make all transactions traceable, and even transfers the liability to the PSP regarding fraud.	With Okay we can even store screenshots of what the end user exactly saw during the transaction verification. We can help you prove that the user was not fooled by malware!
☐ All parts of the security solution are **audited and documented.**	PSD2 §3 states that "The implementation of the security measures referred to in Article 1 shall be documented, periodically tested, evaluated and audited in accordance with the applicable legal framework".	We have performed audits with third party experts, SRC GMBH from Germany, and PROSA Security from Norway.
☐ You're protected against **innovative new forms of malware** directly targeting your app.	The RTS requires that the solution should allow for protecting against new threats to the security of electronic payments.	Our fundamental strategy in designing the Okay solution is that "no device is secure". We focus only on the sensitive part of your app, allowing us to implement much more advanced security than other solutions.

https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32018R0389
https://ec.europa.eu/info/law/payment-services-psd-2-directive-eu-2015-2366_en

okay

Obfuscation

Malware resistance through obfuscation. Just-in-time transfer of obfuscated executable code and server-side verification.

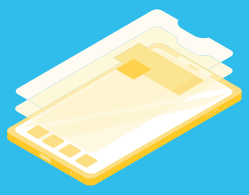


Honeypots

Tamper resistance through honeypots & multiple integrity tests.

Continuous Encryption

Encrypted storage which can only be decrypted with a one-time key from a secure server before immediate re-encryption.



Floating Windows

Active overlay detection mechanisms for floating windows and the like.

Robot Detection

User verification through accelerometer readings.

Server-Side Screenshots

Passive overlay detection through screenshots analyzed on the server-side.