Mobile E-Signing

Paperless Contracting in Mobile Sales and Service Delivery

NAMIRIAL GmbH

Legal Office: Seilerstätte 16, 1010 Wien, Austria
Main Office: Haider Straße 23, 4025 Ansfelden | Phone: +43-7229-88060 | www.xyzmo.com
Fiscalnumber 09 258/9720 | VAT-ID: ATU70125036

Abstract

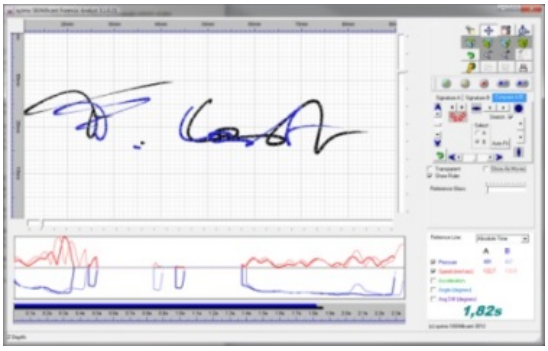
In today's competitive business climate, it is essential to seek cost-cutting possibilities to improve operational efficiency and to pay attention to customer interests and demands to improve the bottom line. Printing documents just to capture a customer signature is not only completely outdated in today's tablet-pervasive everyday life but is also a great waste of time and money. More than that, paper handling is very time-consuming for mobile sales and service personnel and thus reduces the possibilities for efficient customer communication, which in turn limits upsell and cross-sell opportunities.

Modern e-signature-based digital document processes are now geared up to remedy the situation, as they are able to close the final gap in the attempt to go fully paperless even when meeting clients on the go. This white paper looks at the specific requirements for e-signature software in typical mobile sales and service use cases that can be found in insurance sales or industry service organizations.

First, this white paper helps you to select the most appropriate signing device, deployment model, and document format. Then we take a deeper look at important security aspects. After discussing the best architectural choices for a fast and seamless integration into your environment, we look at all the aspects that are important specifically to mobile scenarios, in which you will also see that e-signing is much more than simply signing digital documents—it's about productivity. Finally, we also illustrate the end-to-end business processes that real-customers have implemented for mobile e-signing with SIGNificant.

Table of Contents

1 Selecting the Right Methodology



Signing digital documents with a handwritten signature that is forensically identifiable is clearly the preferred choice in a mobile sales or service delivery scenario. Although there are other biometric technologies available (e.g., finger-print, iris scan, etc.), handwritten signatures that can be authenticated like on paper by a graphologist (based on capture data such as speed, acceleration, pressure, etc.) and that are uniquely bound to a certain document have finally emerged as the de facto industry standard for e-signing digital documents in a mobile scenario. The key reason for this is likely that consumers culturally understand that the act of signing by hand is “significant”—and if all steps are within their control, intuitive, and visible, as they are on paper, then courts support upholding this intent more strongly than almost any other method (particularly in non-U.S. jurisdictions). A handwritten signature on a document is accepted world-wide as an indication of legal or social contractual agreement. This will not change for a long time, especially when in-person signing is required.

As the signatory signs in an in-person meeting on the mobile device of your sales or service delivery employee, you have the e-signing environment and thus the proper capturing of handwritten signatures and their associated biometric data under full control. For this use case, installing native applications provides the best user experience and enables the capture of biometric data properly.[1] HTML5 applications that do not require upfront installation are an option as well, but by design, they only deliver an image of the capture signature with no biometric data, and they make the sales/service employee 100% dependent on online connectivity.

There are quite a few different biometric e-signature technologies for mobile sales and service delivery available on the market. They can mainly be differentiated into the following three areas:

- Signing device
- Deployment model
- Document format

1.1 Signing device

Portability is obviously the key requirement for signature-capturing devices in a mobile scenario. The easiest is, of course, directly signing on your mobile device—as is possible on touch-screen devices such as tablets or phablets. If you want to

[1] The browser layer of an HTML5 application cannot be fully controlled and thus does not allow proper recording of time-based biometric signature data. Pens with pressure are also not supported in HTML5.

sign with a traditional notebook without a touch-screen, you can use an external device—a signature pad—for signature capturing.

Tablets/Phablets with a native digital pen

Tablets are perfect for mobile e-signing, as they give you the closest user experience to real paper as you read, edit, and sign the document directly on the screen. Ideally, you can do everything you can do on paper—just with a digital pen. If the tablet comes with a pen that:

- Provides palm protection (you can touch the screen with your palm while writing),
- Has a natural shape with a small/thin pen tip enabling you to write fine lines, and
- Has a not-too-slippery surface,

you get a writing experience that is close to paper—which will even allow non-experienced users to sign properly if they do it for the first time.

The question of how important it is that the digital pen is able to record pressure information or not can be controversial. You may get a higher legal evidential weight when signing with a pressure pen—but a graphologist doesn't really need pressure values to offer an expert opinion.[2]

Tablets/Phablets with a capacitive stylus

If your tablet of choice does not come with a native digital pen (e.g., an iPad), you still can use it very well for e-signing. Although signing with the finger technically works, it is recommend to use a capacitive style, as it will provide the user with a better writing experience because nobody is really used to signing with a finger. Of course, the writing experience cannot be as good as with a native digital pen—but it is better than you may think. You simply write a bit larger and slower.

Alternatively, you may use third-party pens that provide one of the aforementioned key benefits:

- Palm protection
- Thin pen tip
- Pressure recording
-

Among those benefits, good palm protection (you can rest your palm on the device while writing) is the most convenient feature for the signer, as many signers tend to touch the device surface with the palm while signing, which interferes with capturing the signature.

[2] See expert opinion by Dr. Caspart.

Laptop computer with a signature pad

If you are using a laptop computer without a touch-screen for document display and editing, you can use a peripheral device to capture a handwritten signature and its biometric data. Special-purpose devices—mobile signature pads—from vendors such as Wacom are best suited to provide a good signing experience and portability. To prevent sniffing of the captured biometric data from the signature tablet, security mechanisms range from encrypting the communication between the signature pad and the computer to an end-to-end encryption of the signature data on the pad itself.

Laptop computer with a connected smartphone

If using a special-purpose signature pad is not an option or desired, e.g., because you are working with an independent sales channel, you can use a smartphone as a signature-capturing device instead. For a true mobile solution, the connection between the laptop and the smartphone has to work without an Internet connection—e.g., via Bluetooth or WI-FI directly. When you are ready to sign a document, secure communication between the smartphone and the host computer is established, and a native app turns the smartphone into a signature-capturing device.

1.2 Deployment model

You can either run a standalone e-signing application on a mobile computer, or you can choose a client/server approach (see chapter 3.2 for pros and cons). Regardless of which option you choose, offline support is absolutely possible (see chapter 4.1).

If you prefer a client/server architecture, you need to choose the deployment model of your e-signature server infrastructure. It is possible either to run it in the public or private cloud and consume it through a SaaS model or to deploy and run it on your own premises. Whereas the cloud model is faster and easier to set up, it typically provides only limited options to define where your servers and data should be located, and you make yourself 100% dependent regarding availability and security from the provider. Therefore, there are still good reasons—data protection and residency issues are just two of the obvious examples—to deploy on-premises behind a firewall, providing maximum control over data and systems.[3] Keep in mind also that starting with a cloud service is much easier than getting out. Make sure that you can decide at any time to move the applications back to your data center if required.

[3] http://www.zdnet.com/how-one-judge-single-handedly-killed-trust-in-the-us-technology-industry-7000032257/

1.3 Document format

According to Gartner Research (Publication ID Number: G00159721), the best **document format** is self-contained, so it includes the content to be signed, the signature, and the metadata to make it searchable, and it stores the information needed for proof in addition to the signature—the date, time, and consent. It should also only require a freely and ubiquitously available reader to show the document in its **originally** archived form.

Other than proprietary document formats and document databases, the open Portable Document Format (PDF) fulfills all these requirements. PDF not only is an open standard defined in ISO 32000-1:2008 but also comes in a variant designed for long-term activation defined as a PDF/A in ISO 19005-1:2005. Additionally, digital signatures are well defined within the PDF itself (Adobe PDF Reference PDF 32000-1:2008 12.8.3.3 PKCS#7 Signatures—as used in ISO 32000), meaning that every standard compliant viewing application, such as Adobe Acrobat Reader, correctly shows digitally signed PDFs. Therefore, a PDF or PFD/A file is the perfect analogue to paper in the digital world for archiving signed document originals. All signatures and their cryptographic information should be embedded into the signed PDF. It should concern you if you need to be a customer of a certain e-signature provider or return to their website just to check the validity of documents.

2 Security Aspects

Electronically signed documents will be in future your legally bound originals. Thus, security has to be bulletproof; otherwise, the digital originals would become worthless. Therefore, security aspects are a major topic. The most important aspects are pointed out in this chapter. For more detailed information, please ask for the SIGNificant security whitepaper.

2.1 Authenticity protection

Protecting the authenticity of a signature and its binding to a certain document and position within a document is core to all security aspects of e-signing. It simply must not be possible for an attacker to access and copy the signature data of one document and paste it somewhere else—whether it be within the same document or into a new document. Thus, secure encryption of the raw data—the captured biometric signature—together with the document fingerprint (= hash value) is critical.

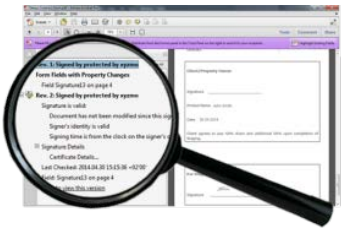
Here, asymmetrical encryption using a hybrid RSA/AES encryption algorithm is viewed generally safe and has been emerged as the de facto industry standard. Today, nearly all important signature capture devices can perform these asymmetric encryption operations directly on the devices themselves, thus efficiently preventing wiretapping of the biometric signature data.

Naturally, proving the document's signature binding should also not depend on the availability of the signature-capturing device on which the signature was captured, because signed documents have a much longer time span than the those devices do.

2.2 Integrity protection

Once a document is signed, it is essential that it can easily be determined whether the signed document is still original or whether it has been altered after the signature has been applied. This kind of integrity analysis must be easily available to everyone who is viewing/reading the signed document; otherwise, forging the content of signed documents is as easy as it is on paper.

2.3 Audit trail



Audit trails should track what happened with a specific document in what order, at what time, and where. A self-contained document with all signatures and digital certificates, including its audit trail, can reside in any storage system and does not need to be kept in a proprietary vault.

2.4 Limiting access to signed documents

In contrast to paper, digital files can be easily copied without losing any of their characteristics. If a digital file is an original, a digital copy of it creates another valid original. In case you want to limit access to an original signed document for security reasons, you must make sure that the e-signing solution does not simply distribute the original file to all decentralized signing stations—which would significantly increase the complexity of securing access to the signed original.

3 Architectural Choice for a Fast and Seamless Integration

An e-signing application typically consists of a front-end and a back-end component. While the front-end software manages all user interactions, the back-end software processes the document and takes care of its integration into the overall document workflow.

The front-end software component naturally runs on a front-office computing device, which can be either:

- A traditional laptop/notebook, which uses an external signature screen or signature pad to capture a handwritten signature, or
- A tablet computer.

Typically, the front-end part is either a standalone pre-packaged GUI application or an SDK that can be seamlessly integrated into an existing client application.

The back-end software component can run either locally together with the front-end component inside the same application/on the same computer or be split off into a separate server application, which means that the e-singing application is distributed over a client and a server.

In the following sections, we will look at the pros and cons for each option.

3.1 Standalone GUI App or SDK

If you require fast and cost-efficient deployment with ready-to-go graphical user interfaces, a standalone GUI application is typically the best and most cost-efficient choice. If done well, this option still allows the easy customization of color schemes, logos, etc., to customer requirements.

If you, however, require seamless integration into an existing application (without a UI context switch) then the SDK approach will be the right one. Here you can manage the detailed user experience and all GUI elements through your own coding. Powerful SDKs allow much more than the simple integration of core functionalities and provide a complete adaptable user interface with a framework to seamlessly integrate it.

But you shouldn’t underestimate the efforts that need to be invested in making a working solution from an SDK, resulting in much higher costs when programming your own solution in comparison to customizing a standard solution. Starting with a ready-to-go solution can be critically faster, as typically, the customization process is done within a few days. Furthermore, there are additional costs that should be taken into consideration, such as maintenance and future developments.

Another important aspect, especially when integrating electronic signature solutions, is that, when using an SDK, your IT department will need to manage all security issues independently. For instance, it must protect customers’ handwritten signatures from being transferred to an unauthorized document or stored somewhere in raw format, which is an enormous burden on the IT department. This includes efforts to avoid misuse and careless coding by the company’s own employees, making the task of providing a sufficient level of security tougher. The fact is that, even if all security actions are implemented, this issue is usually not easily explained to end customers and third parties, as theoretically, the company has the opportunity to misuse this sensitive data. In contrast, in the scenario of a ready-to-go solution, you can easily prove to end customers and, if necessary, to a court in case of a legal dispute, that it had no opportunity to manipulate signatures.

At the end of the day, you must decide what the best solution is for your purposes, but remember that, in contrast to what may be expected at first glance, a ready-to-go solution might be cheaper and deployed faster than an SDK-based approach.

3.2 Signing on the server—pros and cons

Even when opting for an on premise deployment model versus a cloud service, in many scenarios, a centralized server-based approach for the back-end software component running from your own data center has many advantages over a pure desktop-based approach:

- If existing systems for document creation, workflow management, and document archiving are also server-based, the server-side integration is simply much easier.
- The PDF document is only stored in the secure data center and not automatically distributed to the clients, where access to the signed original cannot be securely managed.
- A rich server-side audit trail providing additional process evidence
- A server provides a single point and type of integration for all the different client options:
    - signature pads—managed by a web application or local SDK to be integrated in custom-rich client application
    - signature screens—controlled by a local Kiosk SDK that you can also integrate easily into a your own Web application
    - smartphones—that run a small signature capture app that connects with a Web application to view the document
    - tablets—that run native signing clients to display, edit, and sign documents
- Compatibility to additional sales channels—thus reusing the e-singing infrastructure and software integrations already implemented for the POS in a multichannel environment that also includes mobile and online channels

In contrast, purely desktop-/local-based signing approaches are typically preferred if:

- the document to be signed is dynamically created on the client (and cannot be instantiated from a local static document template), which makes the use of a client/server solution while being offline impossible
- server-side integration is not necessary at all

4 Aspects That Are Important to Mobile Scenarios

The typical end-to-end business process for e-signing on the go differs from other use cases such as, e.g., stationary POS. Consequently, you are faced with requirements that are unique to this use case. The most important ones are listed below.

4.1 Work with documents offline—without any Internet connectivity

In a mobile scenario, it is critical that you can complete your business transactions even in cases where no Internet connection is available. This means that you must be able to read, edit, fill out forms, and sign a contract when you are completely offline—e.g., when you are meeting a client at home. Ideally, you should also be able to create new document instances from templates, automatically pre-fill form fields with data from third-party applications, and manage documents locally without any network connection.

This means that client/server applications must cache the required data locally to support such offline use cases. Of course, if the PDF document itself is created on the local device (e.g., by a third-party application), you will require a standalone e-signing solution to allow complete offline processing (otherwise, you will first require online connectivity to upload the document to the signing server before you can go offline).

4.2 Edit and fill out PDF forms like on paper

Ideally, clients will want to work with digital documents like they are used to doing with paper documents. This means that the e-signing application certainly must allow clients to browse and review multipage documents before they edit and sign them—ideally directly on the signing device.

But you can easily go beyond those basic features, as tablets also allow the editing of documents like you are used to in the paper world. This includes freehand and text annotations, mark-ups, attachments, and filling out machine-readable form fields. Also, the integration of the tablet-based signing solution with the document workflow is key, as you may want to push a pre-filled form document (e.g., a client contract) to a specific tablet device and allow the client to read and update its form field values and then save any update that the client made to the form field values back into your own database.

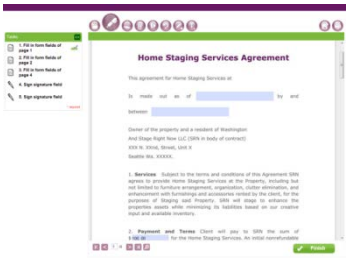
4.3 Add photos taken from the device camera

Mobile devices such as tablets or smartphones are equipped with built-in cameras that are perfect for adding photos to documents, such as pictures from IDs or providing proof of certain situations. The added photo has to belong to the document that will be signed and thus can't be edited without breaking the digital seal of the signature. Ideally, the photo can be added to the document in any size at any given place that can be defined ad-hoc or through a pre-defined template. Alternatively, it should be possible to simply attach it to the PDF document like any other file.

4.4 Add a geolocation to the signature

Sometimes it is important to know the location where certain documents are signed—or it simply adds additional evidential weight. Mobile devices such as tablets or smartphones typically come with a built-in GPS sensor that can be used to include GPS coordinates in the digital signature.

4.5 Avoiding incomplete contracts



Trying to fix ill-signed contracts often is very time consuming and costly because, when you discover the problem, you are typically not with the client anymore. Thus, it is a huge benefit if you can control and govern all steps in the completion and signing process of documents, including filling out form or signature fields, reading pages, accessing scanners or the camera to add attachments such as ID scans, and much more. Ideally, you can specify compulsory or optional tasks depending on the use case and the document, thus giving you the flexibility you need to cover all your business cases.

Additionally, through defining policies that enable or forbid certain actions on or with the document, such as making annotations, saving, e-mailing, or printing documents, you can exercise any further required control over what clients and sales/service employees are allowed to do with your signed documents.

4.6 Integration into the general e-signature platform

It is very beneficial if the mobile signing solution is tightly integrated with the overall e-signing platform, which also covers other channels such as in-house POS or pure remote-online scenarios. Only then can you ensure that integration efforts can be reused and are thus made in the most efficient way.

5 SIGNificant-References

SIGNificant provides an enterprise e-signature platform that allows you to go completely paperless on the go, be it a mobile sales or service scenario. SIGNificant simply provides you with the user interface and tools needed to define an optimal e-signature process and user experience. The platform's building blocks make it easy to pick and choose the best combination of e-sign solutions and signature capturing devices according to customer needs.

To better illustrate how SIGNificant can be applied in different mobile use cases, the following sections outlines real case studies with their end-to-end business process that has been implemented.

5.1 Swiss Life Select

Use case:

- Applications for financial investments and insurance contracts sold through their direct sales force.

Deployed products:

- Signing application: SIGNificant Server with iPad App, Android app and Web Signing Interface.
- Signature capturing hardware: iPads, Android, Surface Pro tablets and Wacom signature pads.

End-to-end business process:

1. The financial advisor loads the contracts that they want to get signed on their tablet through the Swiss Life Select sales portal.
2. The advisor visits the client, where potentially no 3G/Internet connection is available.
3. The client reads the financial service contract and fills out the necessary form fields in a machine readable way (typewriter, checkboxes) directly on the tablet or Notebook
4. The client finally signs the filled out insurance or investment contract directly on the tablet or using a signature pad connected to their Notebook.
5. As soon the tablet gets back Internet connection, the e-signing app synchronizes with SIGNificant Server, which processes and signs the document with the captured biometrical data from the handwritten signature and then digitally seals it with the Swiss Life Select signing certificate.
6. Swiss Life Select securely archives the biometrically signed original PDF documents
7. The client can either access a flattened and digitally signed copy of the original document or have it e-mailed.

5.2 Nürnberger Insurance

Use case:

- Applications for life and non-life insurances sold through their independent channel sales.

Deployed products:

- Signing application: SIGNificant Server with iPad app, Android app and Web Signing Interface.
- Signature capturing hardware: iPads, Android tablets and StepOver signature pads.

End-to-end business process:

1. The broker creates one or more insurance application documents that should be signed electronically through entering the relevant client data on the Nürnberger sales channel portal
2. The broker loads the insurance application contracts that they want to get signed on their tablet through the Nürnberger sales channel portal.
3. The broker visits the client(s), where potentially no 3G/Internet connection is available.
4. The client finally signs the filled out insurance contract.
5. As soon the tablet gets back Internet connection, the e-signing app synchronizes with SIGNificant Server, which processes and signs the document with the captured biometrical data from the handwritten signature and then digitally seals it with the Nürnberger signing certificate.
6. Nürnberger securely archives the biometrically signed original PDF documents.

7. The client can either access a flattened and digitally signed copy of the original document or have it emailed.

5.3 Lower Austrian Insurance (Niederösterreichische)

Use case:

- Applications for life and non-life insurances sold through their direct sales.

Deployed products:

- Signing application: SIGNificant Client running on Windows laptop computer.
- Signature capturing hardware: Wacom STU-500 signature pads.

End-to-end business process:

1. The sales person loads the required insurance application contracts on their laptop computer.
2. The sales person visits the client, where potentially no 3G/Internet connection is available.
3. The client reads the insurance contract and fills out the necessary form fields in a machine readable way (typewriter, checkboxes) directly on the laptop computer.
4. The client finally signs the filled out insurance contract with the Wacom STU-500 signature pad.
5. SIGNificant Client kicks of a post-processing program that securely archives the signed insurance contract.

The client can either access a copy of the digitally signed original document or have it e-mailed.