ONTASK
by accusoft

Build or Buy? Adding Digital Signature and Workflow Capabilities Into Your Application

There was a time when the only way to sign electronic documents was to print them on physical sheets of paper, sign them by hand, and then scan them back into electronic form. Thankfully for everyone, those situations are increasingly becoming a thing of the past with the development of user-friendly applications that deploy eSignature technology.

Developers faced with implementing these signature features into their software are quickly confronted with a dilemma. Should they create their own eSignature capabilities in-house or would it be more efficient to purchase them from a third-party developer? While there are benefits to both approaches, most developers are now turning to existing third-party integrations that take advantage of proven technology and allow them to meet their project deadlines.

Challenges with Building eSignature Features

Development teams often assume it would be better to build their software's eSignature features from the ground up. For many designers, creating signature tools that are customized to meet an application's exact use case sounds like it would improve the user experience while minimizing potential complications caused by external solutions. Since the team would maintain control over the code involved, they could also ensure that the integration with other systems would remain as clean as possible.

Unfortunately, developing those features in practice doesn't typically work this way. Truly robust eSignature solutions require specialized expertise that software teams often lack. When creating a new application, most of the engineering team's focus is understandably directed toward efforts to deliver innovative new features and provide a compelling user experience. Since eSignature capabilities are largely a "solved" solution, the design team rarely has much experience with the pitfalls involved and probably hasn't spent much time thinking about how to reinvent that proverbial wheel.

With that in mind, here are a few reasons why building an eSignature solution from the ground up can be more of a headache than it's worth.

Strategic Workforce Planning

Time is a precious commodity on software development projects. Teams are forced to make difficult choices about where to focus their limited resources to avoid falling behind schedule. This is challenging enough when developers are trying to design, test, and refine new features. Building a unique eSignature solution from scratch can take months to get right if a development team isn't already well-versed in the technical aspects of electronic and digital signatures. Even in a best case scenario in which they do have the right experience and resources available to devote to the project, teams will still need many months to code, test, and implement eSignature capabilities.

Unfortunately, every moment spent working on features that could be easily implemented through a third-party solution is time not being spent on developing the core features that will help an application stand out in a crowded and competitive market. This can quickly contribute to lengthy delays that push back product releases.

They're Expensive to Develop

Designing and deploying a fully-featured eSignature platform is a costly undertaking. Development teams need to build out new backend infrastructure to store and manage signatures securely, implement APIs that connect to various integrations, and create a frontend user interface that works equally well across all devices and screen types. Each of these components requires substantial developer hours to build, to say nothing of the additional hardware or cloud resources that may need to be provisioned to support eSignature capabilities.

Backend and API development typically costs between $15,000 and $32,000 on top of other project costs, depending upon the scale of the project. Designing a browser-based or mobile frontend interface is only slightly less expensive, ranging anywhere from $10,000 to $25,000. This, of course, assumes everything runs smoothly along the way, with no unexpected issues or complications. Recurring problems and delays can cause projects to go over budget, adding even more to total development costs.

They Create Extra Security Risks

One reason why eSignatures features are so difficult to build from the ground up is the amount of security features required to protect the integrity of signatures. To be legally binding and admissible in court, eSignatures must have an audit trail that tracks and time-stamps them. They also need to provide defensible proof that they could be accessed, reviewed, and signed by all parties involved in the process. Personally identifiable information (PHI) needs to be protected by SSL encryption and stored on a system using highly secure servers. Implementing the technical aspects of an eSignature is one thing, but building out the expertise and infrastructure to keep documents private and secure is quite another.

They Need to Meet Compliance Standards

As part of creating a secure eSignature platform, developers must meet a variety of compliance standards governing the way data is handled. At the very minimum, that means undergoing audits to ensure compliance with recognized standards like SOC 2, CCPA, HIPAA, and FERPA, which can be an expensive and time-consuming process. To prepare for those audits, organizations frequently need to develop and roll out complex security standards and practices. Compliance certifications must be maintained after the audits are complete with a combination of operational monitoring and continuing education. Many startups and small software developers simply don’t have the budget to manage all the legal requirements surrounding compliance readiness.

A HIPAA gap assessment, for example, can cost up to $30,000, while a full-scale audit can run anywhere from $20,000 to $50,000. Many eSignature applications will also require more than one compliance audit. In this case, the cost of HIPAA compliance may need to be added to the cost of a SOC 2 audit, which can be as much as $20,000 for a SOC 2 Type 1 audit and $60,000 for a SOC 2 Type 2 audit. And that doesn’t take into account the costs of preparing for the audit itself. A typical SOC 2 readiness assessment averages about $15,000. The prospect of adding $175,000 in compliance costs to any project is enough to make a software developer think twice about taking on the burden and expense of managing those requirements in-house.

They Require Ongoing Support

Security and compliance controls aren’t one-time considerations that can be taken for granted after being put in place. It takes the right combination of people and processes to manage them over time to ensure that applications continue to meet the latest security requirements. This can also apply to eSignature technology itself if regulatory changes alter their legal status. The last thing a developer wants is to have their eSignature capabilities suddenly become outdated or lose their legally binding status. Without a dedicated team to manage those features and navigate evolving regulatory standards, applications could be putting customer data at risk. Building an in-house eSignature solution also means making an ongoing commitment to maintain the infrastructure that supports it, which can quickly translate into sizable long-term costs.

Benefits of Using an eSignature API

With all the considerations involved with developing an eSignature solution from the ground up, it should not come as a surprise that many software teams decide it makes more sense to integrate a third-party API from an experienced vendor. By adopting an API-based solution, they can add proven, reliable functionality into their application and gain access to more features and flexibility than they could create in-house. More importantly, commercial eSign APIs typically offer seamless integrations into existing platforms that have already been optimized over time.

While there are a lot of benefits to be gained from buying a ready-made solution, most of the advantages fall under three broad categories:

They're Cost Effective

While there are initial and ongoing costs to implementing an eSignature API, those costs are frequently much smaller than the expenses associated with building eSignature capabilities from the ground up. There's no need to hire specialized engineers, invest in complex compliance processes, or set up expensive security audits. Valuable and limited resources can instead be directed toward developing more innovative application features to meet the evolving needs of customers. In order to maximize the cost benefits of an eSignature API, however, it's important to find a provider that offers transparent pricing and contracts. Beware of promotional contracts that give way to complicated and obscure pricing structures after the first year or so of use.

They Integrate Quickly

Since eSignature APIs are built upon proven technology and provide an easy path to integration, they allow developers to add signature features to their applications much faster than it would take to build them in-house. While the timeline for building those capabilities from scratch could take months or more, an API-based solution can be implemented in a matter of hours. That means teams can quickly add eSignature features to their applications without compromising their development schedule. Fewer delays and unexpected problems mean more time for testing and refinement prior to launch, ensuring that software not only gets to market faster, but will be more robust and stable when it gets there.

They're Easier to Manage

Buying an API-based solution for eSignature functionality effectively outsources not only the development of those features, but also the ongoing management of them. Rather than keeping pace with the latest developments in eSignature compliance or technology, companies can instead leave that maintenance to a third-party provider with the expertise and resources to handle them. There's also the benefit of having experienced support available to troubleshoot any eSignature-related challenges, rather than languishing in a backlog of IT tickets. As new features become available, they can also quickly implement them without having to perform any complex development work. This ensures that the solution's eSignature capabilities will remain just as cutting edge as its other features.

Enhancing Flexibility with eSignatures and Digital Signatures

When exploring potential signature solutions for an application, it's important to evaluate their capabilities closely. Some of them offer only electronic signatures, which are legally binding and use standard methods like email, passwords, and multi-factor authentication to verify identity. Many use cases require an additional level of security that only digital signatures can provide. These advanced signatures require an encrypted digital certificate to certify the validity of the document and ensure no tampering has occurred. They also use Adobe Approved Trust List (AATL) certificates, which make it possible for anyone to validate digital signatures on any device at any time. These certificates are the industry standard for PDF-based signatures, and they can only be issued by authorized certification authorities using password-protected hardware devices that cannot export private keys (which eliminates the risk of keys being compromised).

While eSignatures are acceptable for a variety of everyday use cases, some situations call for greater security to protect against potential tampering. Most government and certification authorities require documents to be signed using digital signatures to reduce the risk of fraud and allow them to verify authenticity.

It's important to choose a signature API integration that provides both eSignature and digital signature capabilities. Even if an application only requires electronic signatures, having the ability to implement more secure digital signatures in the future can prevent headaches in the future should new software features call for higher levels of security and verification.

eSignatures vs Digital Signatures

eSignatures

Electronic representation of a wet-inked signature.

eSignatures are used to:

- Show intent to sign
- Gain visibility into who has signed
- Collect legally binding signatures

VS

Digital Signatures

Virtual identifier used to validate and secure a document.

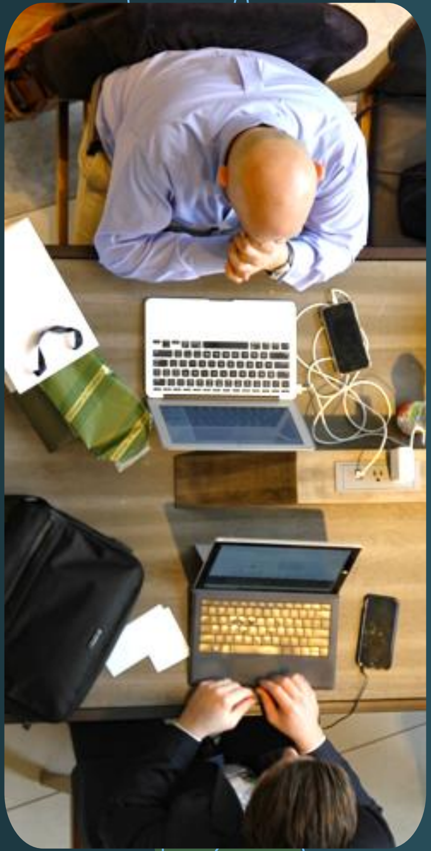
Digital signatures are used to:

- Secure documents with encryption
- Validate user identity using authentication methods
- Tamper-proof documents and avoid changes after signing
- Improve audit trails with timestamps and digital certificates

Why eSignatures Need Workflow Capabilities

Of course, simply gathering eSignatures is only one part of a larger process. After using an eSign API to solve the technical hurdle of requesting and collecting signatures, the application often needs to be able to send them somewhere for processing, review, or storage. Depending upon the eSignature use case, several people may be involved in the process, which would require the application to be able to designate different roles and permissions within a defined workflow.

The need for these workflow capabilities presents development teams with the same build vs buy question that is faced with eSignature tools. Creating workflow features capable of automating repeatable processes using conditional logic is another significant technical hurdle that can easily delay software projects. By implementing an eSign API platform that includes workflow capabilities, developers can save time and build much more complex functionality into their applications.



How Workflows Enhance Application Flexibility

Conditional workflow logic makes it easy to set rules to intelligently automate an existing business process. While an API-powered workflow could be used to digitize a variety of tasks, it's especially useful for eSignatures. Once the signature is gathered, it can be routed to a variety of destinations instantly, such as sending it along to another person for review, adding it to a document that’s being generated, or archiving the data in a system of record.

In many cases, these workflows may need to branch out as part of a more complex process that incorporates multiple tasks that need to be completed. Automated reminders and notifications can also be set to escalate various tasks if no action is taken. This could include reassigning or resending information or bringing more people into the process to ensure everything remains on track.

One of the major advantages of an API-based workflow platform is that they frequently provide a wide range of templates in addition to customizable workflow builders. These templates allow developers to quickly select and implement a proven, secure workflow for routing eSignatures. Not only can they avoid the hassle of building workflow functionality from scratch, but they can also skip the trial and error of designing a conditional process that meets their application and business needs. If the API offers integrations with other systems, the advantages are even greater because the workflow can securely route data to and from those systems without forcing people to manually transfer information between them.

Unlock Your Application's Potential with OnTask API

OnTask's eSign API helps developers to quickly fill gaps within applications thanks to our detailed documentation, extensive code samples, customizable workflows, and robust engineering support. With a simple API call, your software can connect to the powerful OnTask platform to gather secure, compliant eSignatures and route them through a conditional workflow for archival or review.

Instead of spending time and money solving a challenge that's already been solved, development teams incorporate a solution that's fully HIPAA, SOC 2, FERPA, and CCPA compliant, and has an entire team in place dedicated to maintaining the highest levels of security while developing new signature and workflow features. We use 256-bit AES and TLS encryption to fully comply with HIPAA requirements and protect data both at rest and in-transit to keep customer data protected.

When selecting OnTask API integration options, it's important for development teams to think about both their existing workflow needs and their future requirements. Having the ability to implement and maintain multiple workflows from the very beginning can help to avoid potential challenges in the future. The elegance of API-based workflows often leads users to want to use them for a variety of processes, so it's generally a good idea to keep scalability in mind during implementation.

To learn more about how OnTask API can help you cut development costs and accelerate your application's time to market, talk to one of our API experts today.

SCHEDULE DEMO

ONTASK
by accusoft

info@ontask.io | ontask.io