digicert®

Continuous Integration/Continuous Delivery with DigiCert Software Trust Manager

Modern code signing for CI/CD processes

Traditional code signing is where organizations buy a code signing certificate from a Certificate Authority (CA) and take responsibility for creating the keypair and Certificate Signing Request (CSR) locally in their environment. Thus, they are responsible for protecting the private key within their organization.

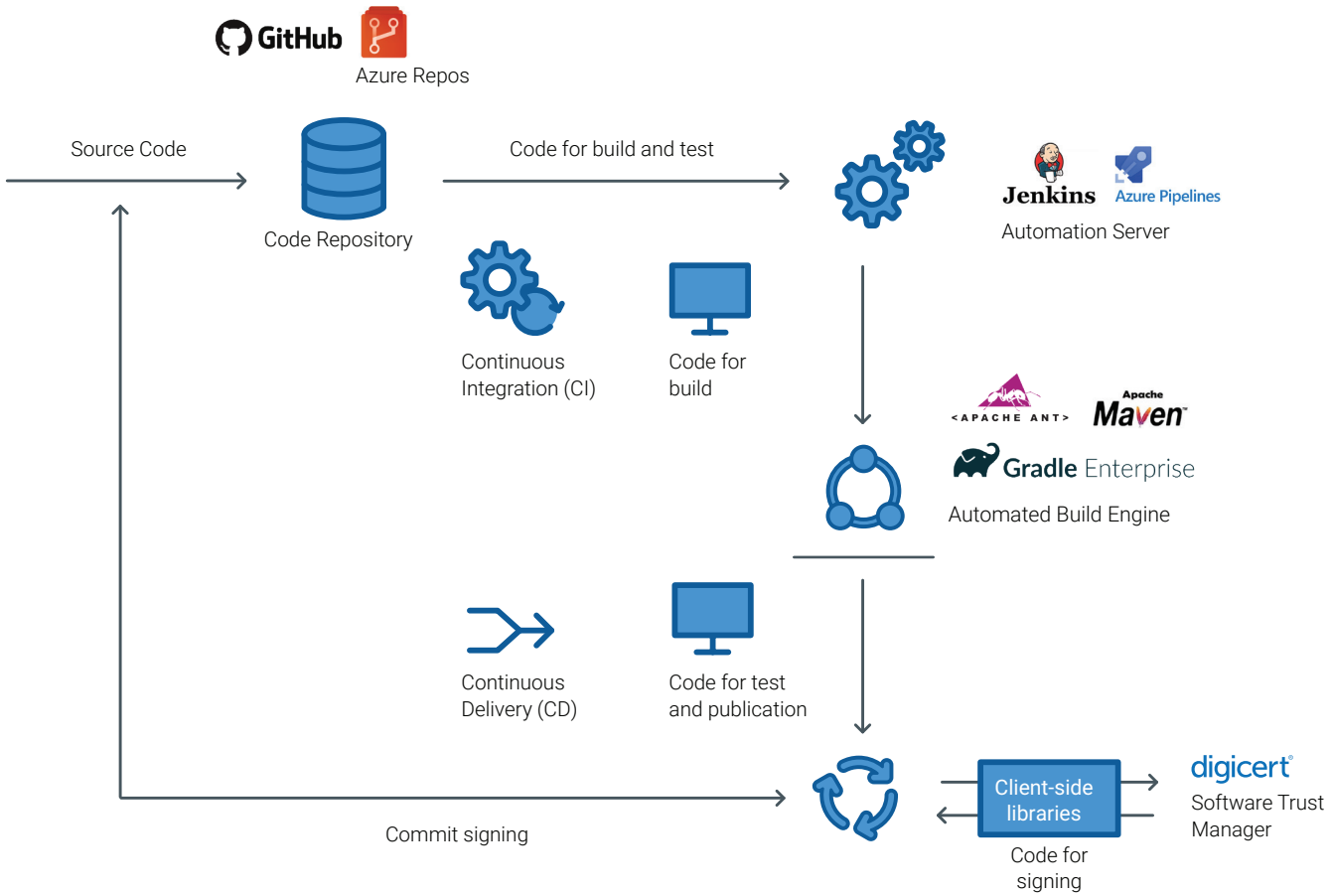
Modern code signing moves away from generating keypairs locally, but instead stores private keys in a central secure location and then manages access to users who are authorized to sign on behalf of the organization.

With traditional code signing, software development teams have to choose between security and productivity. Code is signed quickly, locally, with keys that are shared or kept on desktops or other unprotected devices. These keys are vulnerable to theft or misuse and can be used to sign code and propagate malware seemingly from their organization.

Modern code signing secures signing keys and removes the burden of key protection from developers. Modern code signing can also be integrated seamlessly into developers' CI/CD process so that they can continue to maintain high level of productivity while exercising code signing best practices.

Software Trust Manager drives code signing expediency and efficiency

Code signing assures that your code has not been altered since it was signed. To ensure the integrity of the code, a PKI certificate provides encryption, authentication and identity before Quality Assurance tests and product delivery. The diagram below shows how Software Trust Manager, a modern code signing solution makes it easy for DevOps teams to seamlessly incorporate secure and high-performance code signing with automation build tools in use.



Automated code signing for agile development

- Automate signing by enabling direct integration of Cryptographic Service Provider (CSP) on major CI/CD platforms such as Apache Ant, Apache Maven, Azure DevOps, Gradle, and Jenkins.
- Minimize overhead by leveraging DigiCert client-side libraries (Microsoft KSP, Apple CryptoTokenKit and PKCSII) for scripted integration which can be called from within the CI/CD pipeline in an automated way.
- Adopt repeatable code signing best practices without complex user interactions.

Enhanced code signing security without additional workload from developers

- Remove security burden from developers even when you are maximizing security with multiple key signing models to choose from, including single usage and on-demand keys. These key signing models are aligned with major signing platforms and with your needs.
- Maintain secure access and privileges for signing and administration with permission-based access. In the event of personnel changes, you can adjust access levels quickly and securely from a centralized control panel. A build server can be configured as an API user so that signing requests can be made without the need for human intervention.
- Easily export in-depth reports and logs to track signed code and activities. For a more automated process, you can request reports and audits over APIs.
- Enable future time-based verification with timestamp.

High performance for high-volume development

- Expedite secure signing of large files with "hash" signing which eliminates the need to transfer actual source files to the cloud for signing.
- Increase efficiencies in high-volume development by hash signing all files, including files using public Extended Validation (EV) and Organization Validation (OV) and private code signing, as well as for all major binary types, including Microsoft Authenticode, Java, Android, and Docker.

For more information on Software Trust Manager, contact one of our PKI experts at pki_info@digicert.com or visit www.digicert.com/software-trust-manager