



JS7 JobScheduler Architecture

Security Architecture:

Connections, Access, Operation

**Information for
Interested Parties**



■ Secure Connections

- Network Connections
- Certificate Preparation
- Certificate Deployment

■ Secure Access

- Identity and Access Management
- User Account and Role Management
- Use of Identity Services
- Certificate based Authentication
- FIDO Authentication

■ Secure Operation

- Secure Deployment: Security Level Low
- Secure Deployment: Security Level Medium
- Secure Deployment: Security Level High
- Secure Roll-out

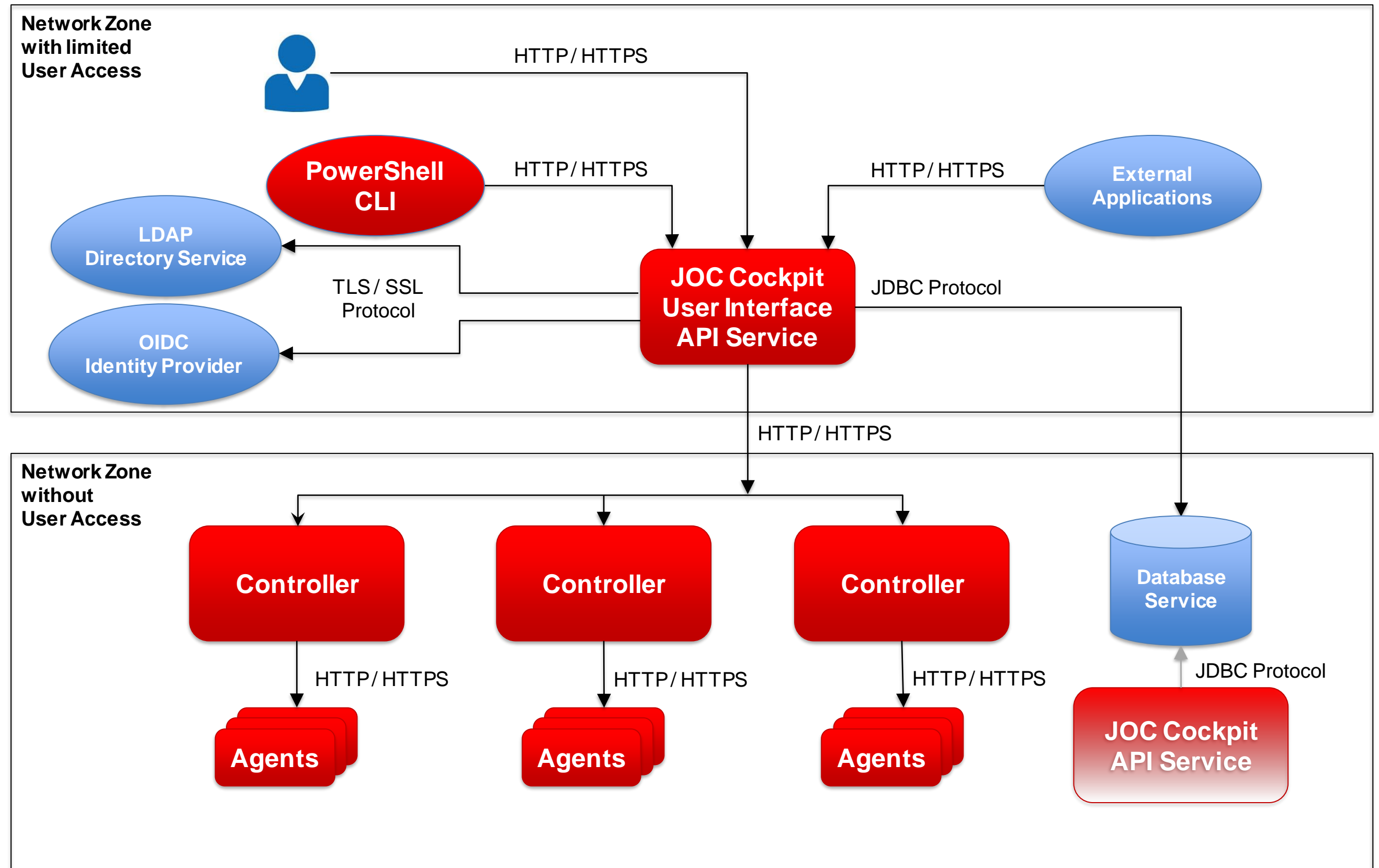
Secure Network Connections

Network Zone with restricted User Access

- Use of HTTPS for any connection to JOC Cockpit
- Access to JOC Cockpit requires authentication
- Access to JOC Cockpit is authenticated by the API Service using TLS/SSL

Network Zone without User Access

- Controller and Agent instances can be operated in a network zone without user access
- Controller instances are accessed exclusively by the JOC Cockpit API Service
- Agent instances are accessed exclusively by Controller instances
- Use of HTTPS for connections with client and server authentication certificates (mutual TLS authentication)



Server Certificates

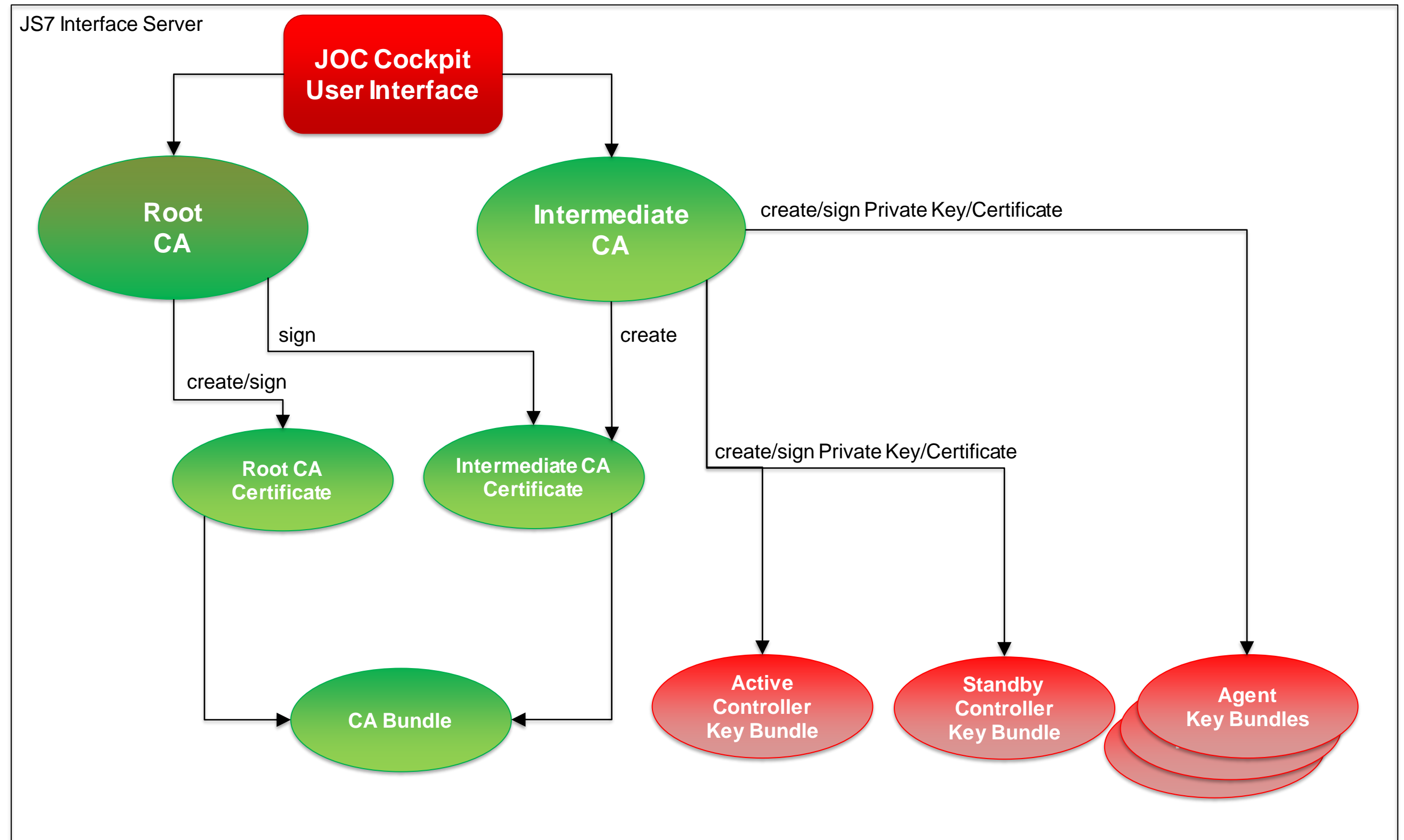
- Server Certificates are required to secure network connection by HTTPS
- Certificates are managed by the user's CA or by the CA provided with JOC Cockpit

Root CA / Intermediate CA

- The Certificate Authority (CA) is used to create the Root CA Certificate and Intermediate CA Certificate
- Both certificates are bundled and made available to Controller and Agent instances

Controller/Agent Certificate

- The Intermediate CA creates and signs the certificates for each Controller and Agent
- The CA Bundle and the instance's Key Bundle are deployed to the respective Controller and Agent

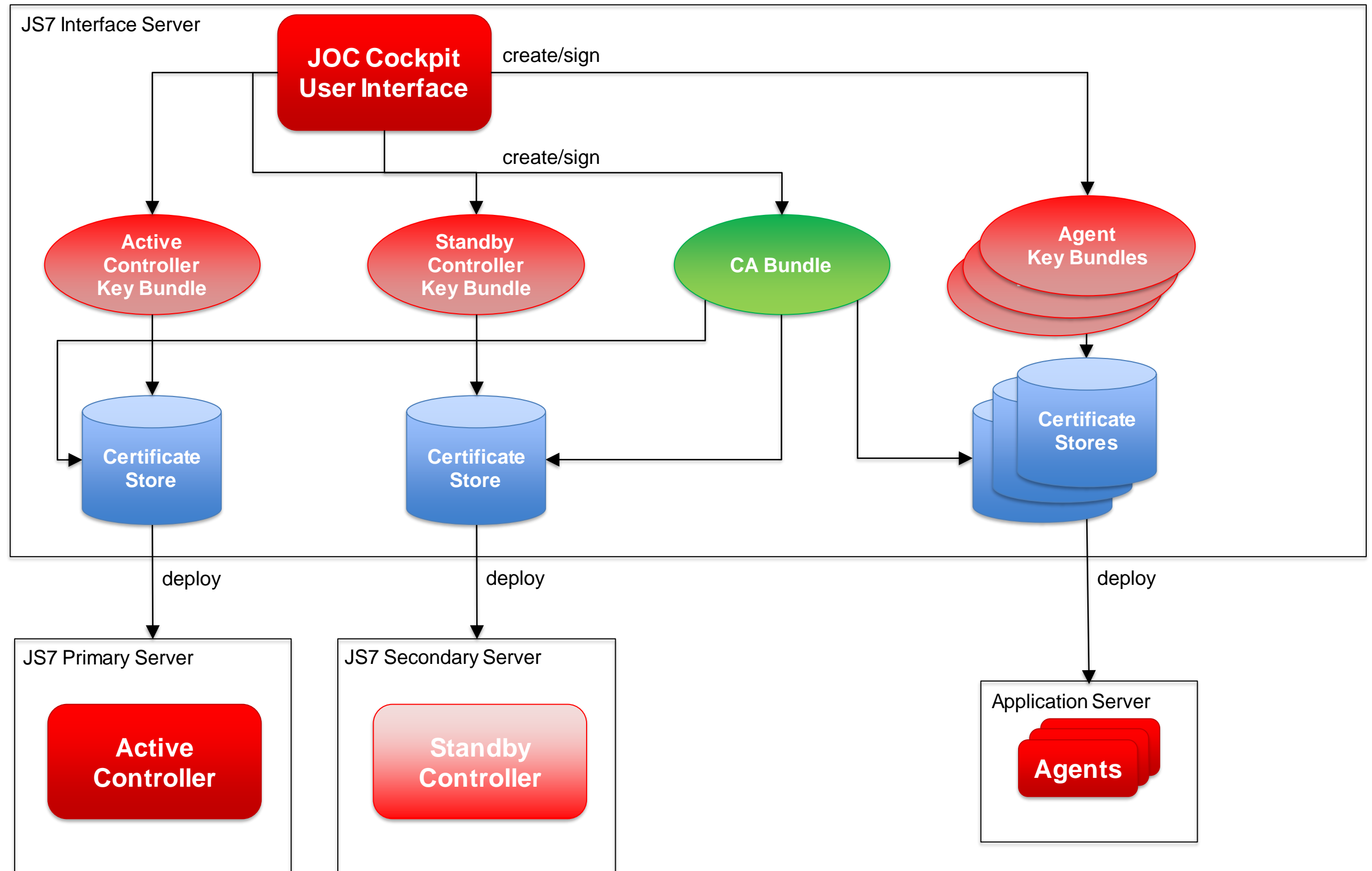


Private Keys / CA Bundle

- The Certificate Authority (CA) adds the Root CA Certificate and Intermediate CA Certificate to a Bundle
- The CA Bundle together with the Private Key is added to a Certificate Store that is managed for each Controller/Agent instance

Deployment

- The Certificate Store is deployed to each Controller/Agent instance
- The transfer of Certificate Stores to Controller/Agent instances is integrated with the user's deployment solution





■ Secure Connections

- Network Connections
- Certificate Preparation
- Certificate Deployment

■ Secure Access

- Identity and Access Management
- User Account and Role Management
- Use of Identity Services
- Certificate based Authentication
- FIDO Authentication

■ Secure Operation

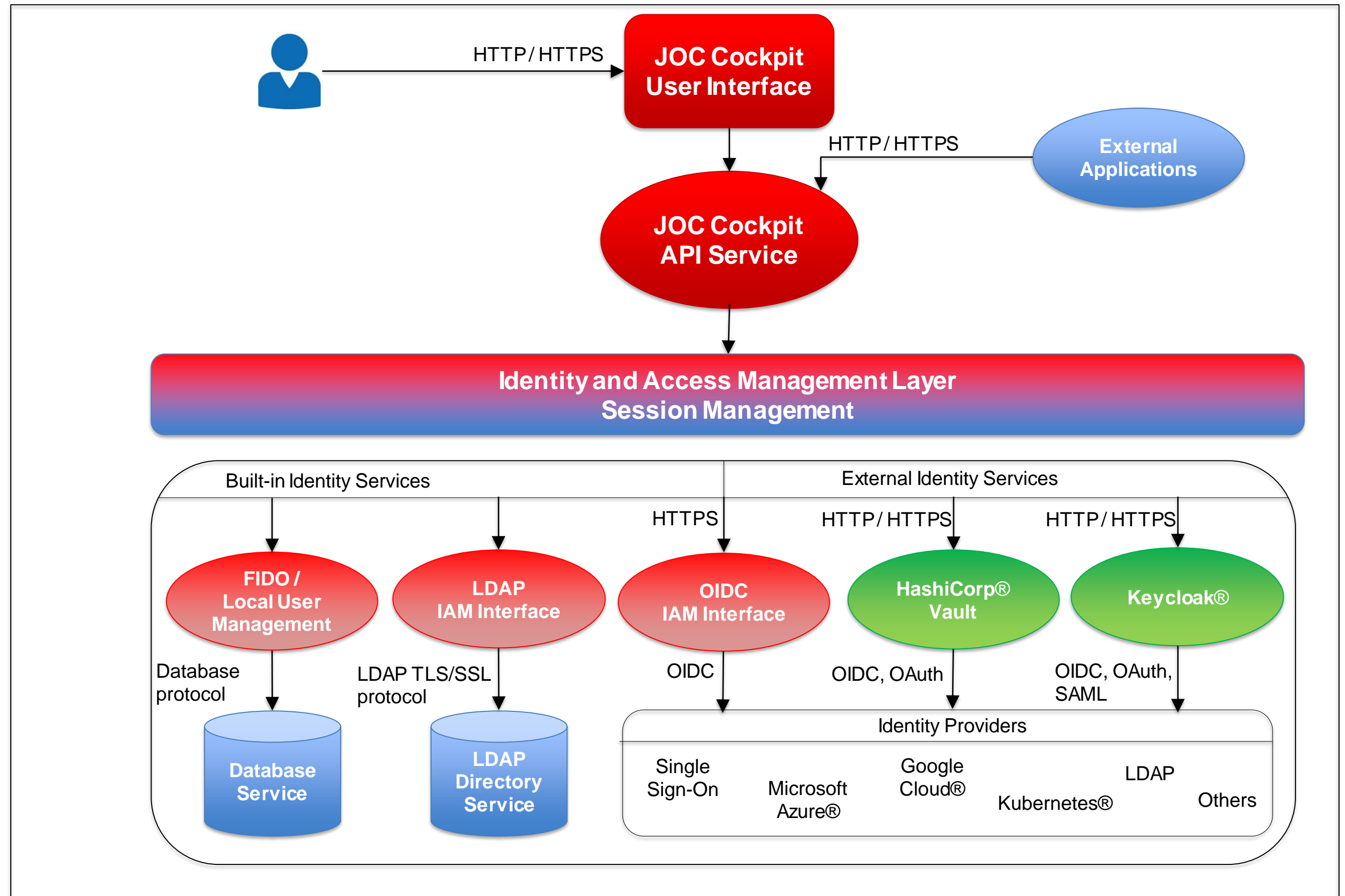
- Secure Deployment: Security Level Low
- Secure Deployment: Security Level Medium
- Secure Deployment: Security Level High
- Secure Roll-out

Built-in Identity Services

- JOC Cockpit offers built-in user management with its Database Service
- JOC Cockpit integrates with LDAP Directory Services such as Microsoft Active Directory® and OpenLDAP.®
- LDAP connections can be configured for TLS and SSL
- JOC Cockpit allows authentication with OIDC based Identity Providers, e.g. Azure
- JOC Cockpit offers FIDO authentication with FIDO2, FIDO U2F and Passkeys

External Identity Services

- HashiCorp® Vault and Keycloak® are integrated by their respective REST API
- JOC Cockpit can manage users and roles locally and by use of an Identity Service
- JOC Cockpit requests and renews access tokens with the Identity Service
- Identity Services manage access to the respective Identity Providers

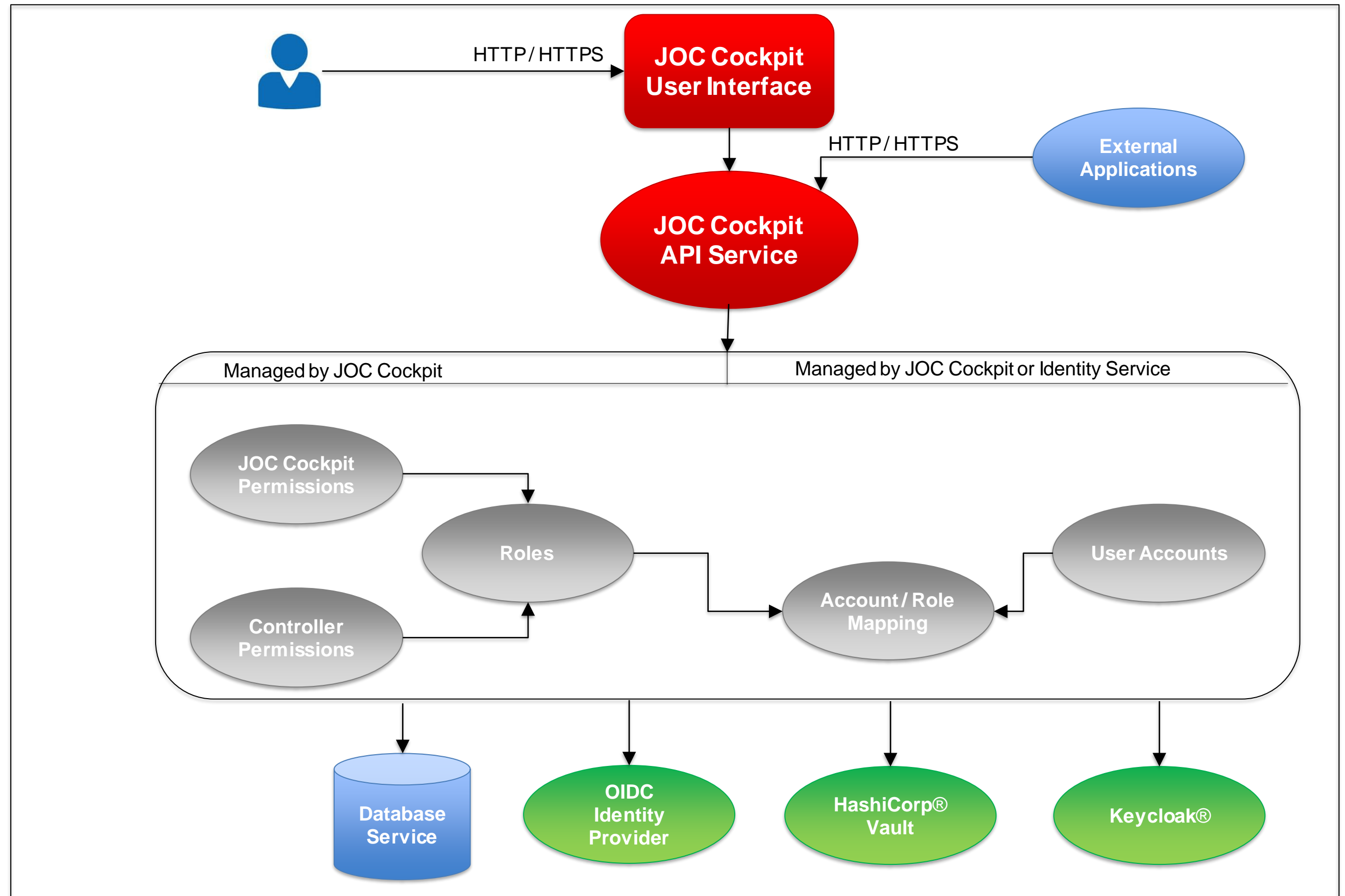


Permissions and Roles

- Permissions for JOC Cockpit and for individual Controllers are managed by the JOC Cockpit API Service
- JOC Cockpit stores permissions and roles with its Database Service

Account and Role Mappings

- User accounts can be managed and stored with the JOC Cockpit database
- Alternatively user accounts can be managed and stored with an Identity Service or OIDC Identity Provider
- JOC Cockpit can be used to populate Identity Services with user accounts and role mappings and it can be used to retrieve role mappings from an Identity Service when a user logs in
- HashiCorp® Vault and Keycloak® are integrated by their respective REST API

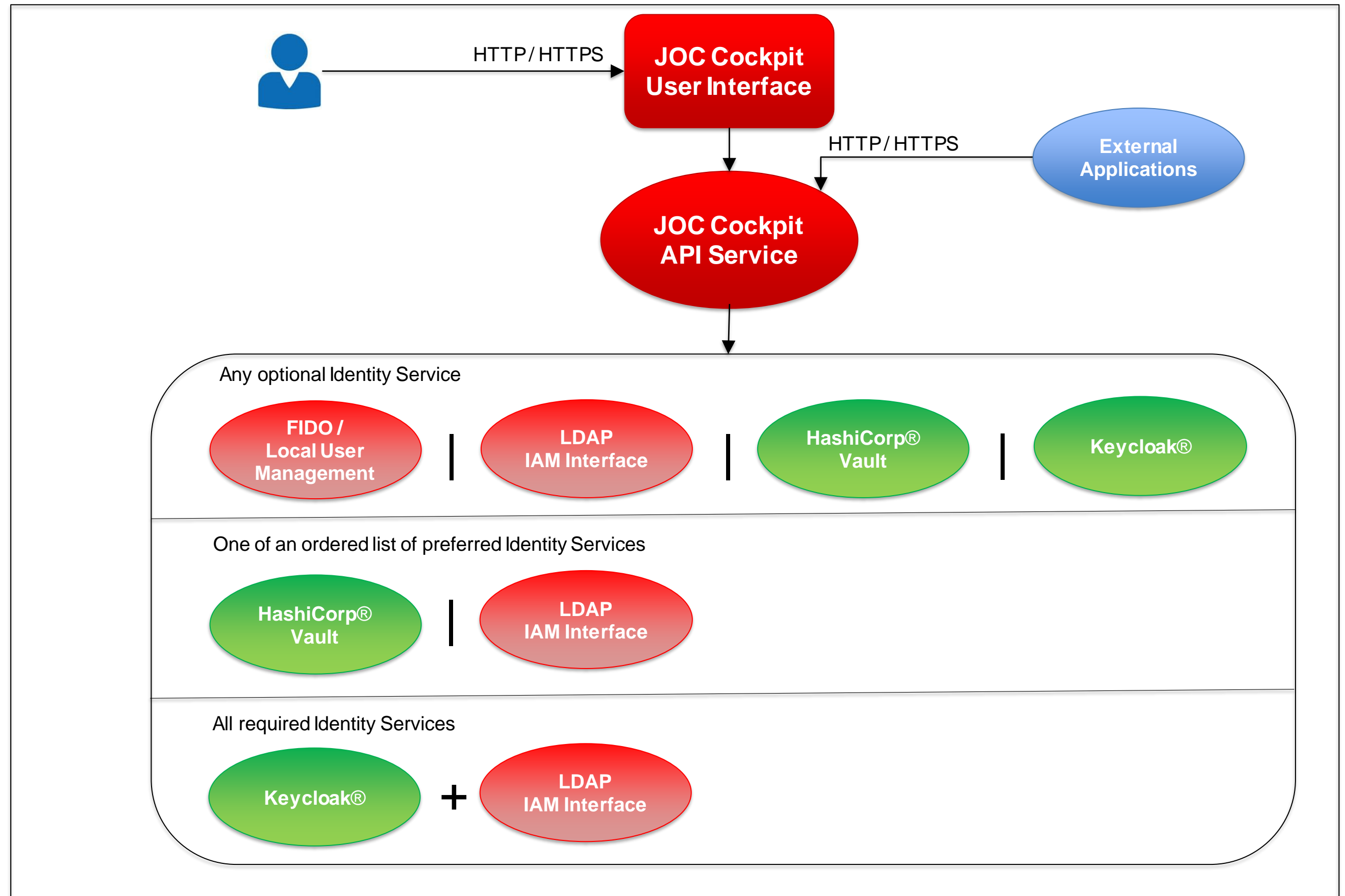


Required Identity Services

- JOC Cockpit offers to specify one or more Identity Services to be required
- This includes to login with all required Identity Services
- A failed login with any required Identity Service results in denial of access to JOC Cockpit
- Role mappings are merged from all required Identity Services

Optional Identity Services

- JOC Cockpit offers to specify any number of optional Identity Services
- With the first successful login to an optional Identity Service the user is logged in and no further Identity Services are consulted
- Identity Services are consulted in the sequence in which they are ordered



Certificate based Authentication

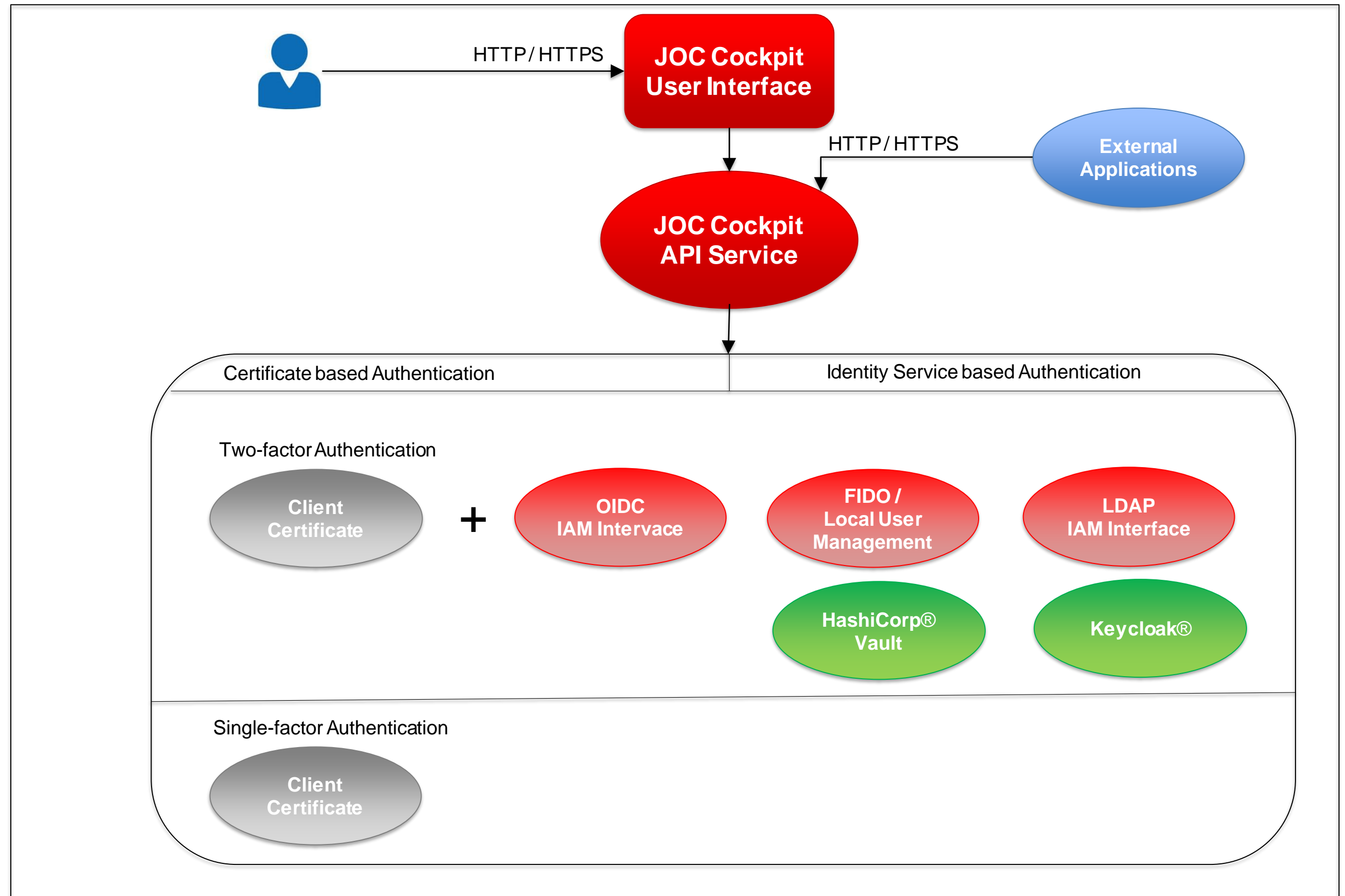
- JOC Cockpit offers use of X.509 Client Authentication Certificates
- Such certificates are used for mutual authentication between JOC Cockpit and Client, e.g. the user browser or external application

Two-factor Authentication

- A certificate is required in addition to credentials that are used with available Identity Services
- If the certificate cannot be validated then the user account is denied access to JOC Cockpit

Single-factor Authentication

- A certificate is accepted as a single factor to authenticate user accounts
- This option is frequently used for automated login by batch processes



FIDO Authentication

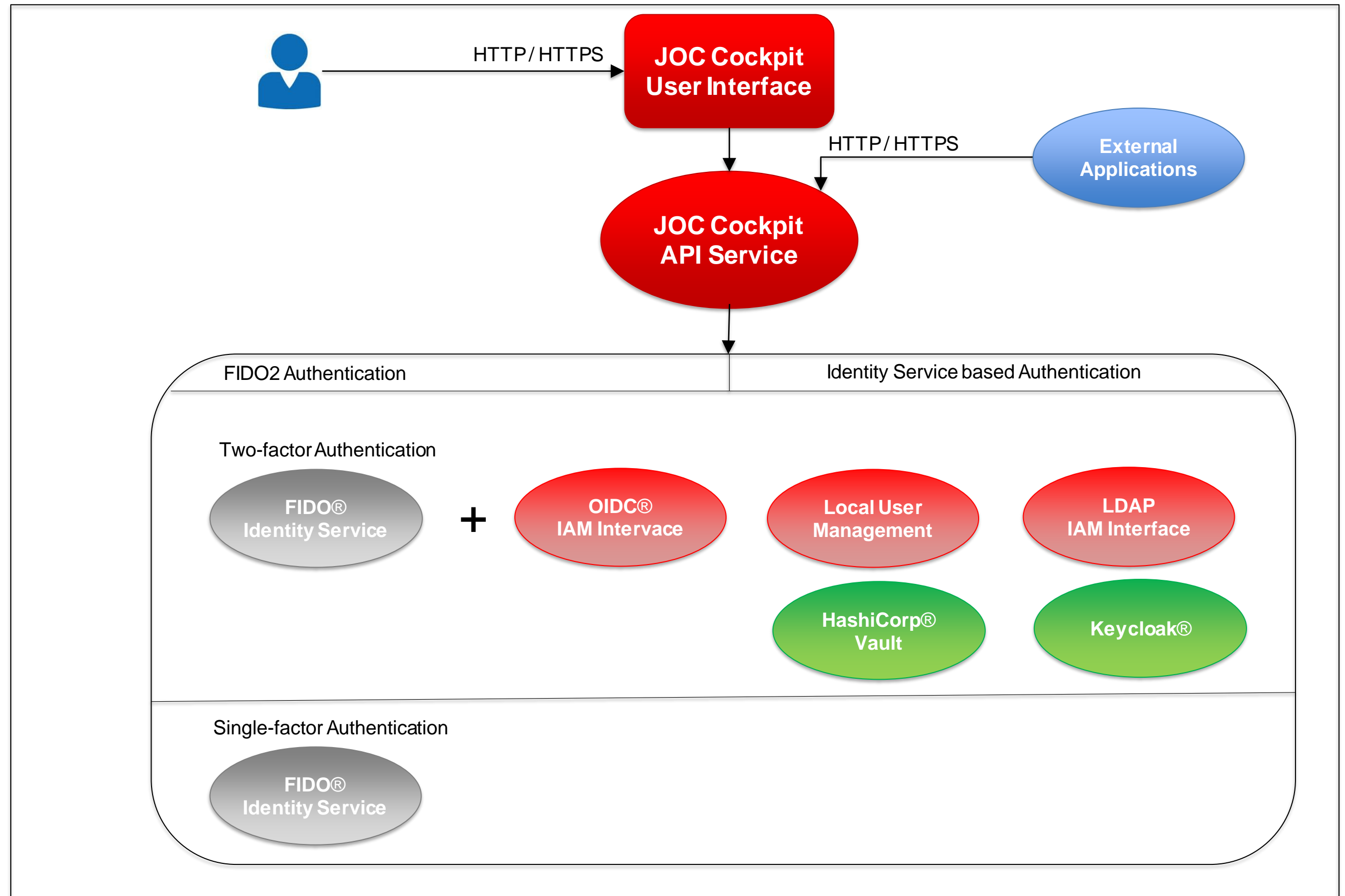
- The JOC Cockpit User Interface implements a FIDO Client for browsers
- The JOC Cockpit API Service implements a FIDO Server (Relying Party)
- Any FIDO compliant Authenticator can be used
- Supported protocols include FIDO2, FIDO U2F, Passkeys

Two-factor Authentication

- FIDO authentication can be used in addition to any other Identity Service
- If FIDO authentication is not successful then the user account is denied access to JOC Cockpit

Single-factor Authentication

- FIDO2 and Passkeys are accepted as a single factor for user authentication





■ Secure Connections

- Network Connections
- Certificate Preparation
- Certificate Deployment

■ Secure Access

- Identity and Access Management
- User Account and Role Management
- Use of Identity Services
- Certificate based Authentication
- FIDO Authentication

■ Secure Operation

- Secure Deployment: Security Level Low
- Secure Deployment: Security Level Medium
- Secure Deployment: Security Level High
- Secure Roll-out

JOC Cockpit Signing

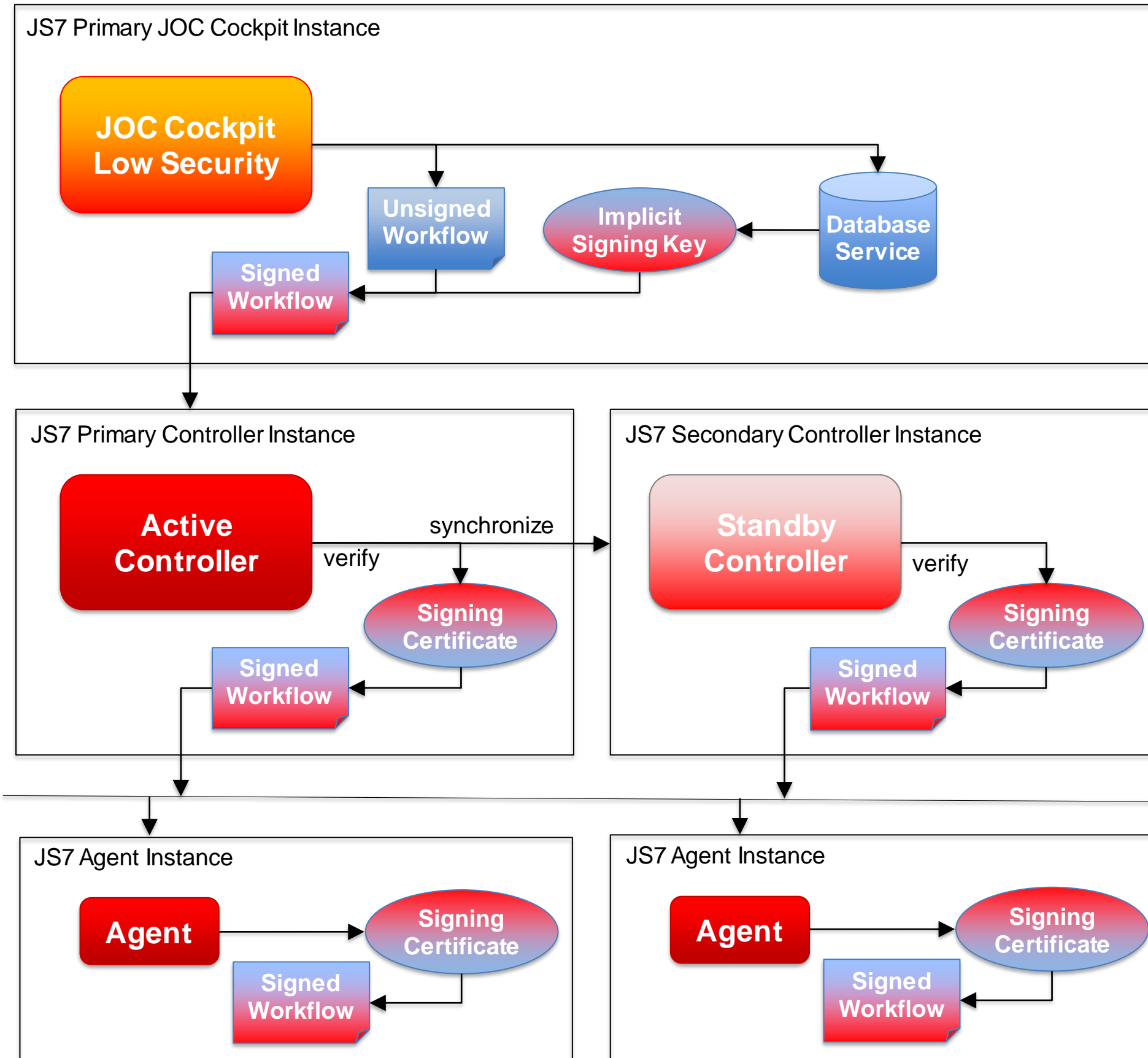
- Workflows are signed for deployment using a Signing Key from the database.
- The signature is deployed with the workflow.

Controller Signature Check

- Workflows are checked if they match the signature provided with certificates available to the Controller and otherwise are denied.
- Similar check is performed by the Standby Controller that requires availability of the same certificates.

Agent Signature Check

- Workflows are checked if they match the signature provided with certificates available to the Agent.
- Workflows are not accepted in case of failed signature checks.

**Implicit Signing**

- Workflows are implicitly signed for deployment
- Deployments by any user make use of the same Signing Key
- Deployment is a single click operation in the user interface

Security Level Low

- JOC Cockpit offers built-in signing operations for workflows
- Signed workflows are not associated with individual users
- The Signing Key might be copied or compromised as it is accessible from the database

JOC Cockpit Signing

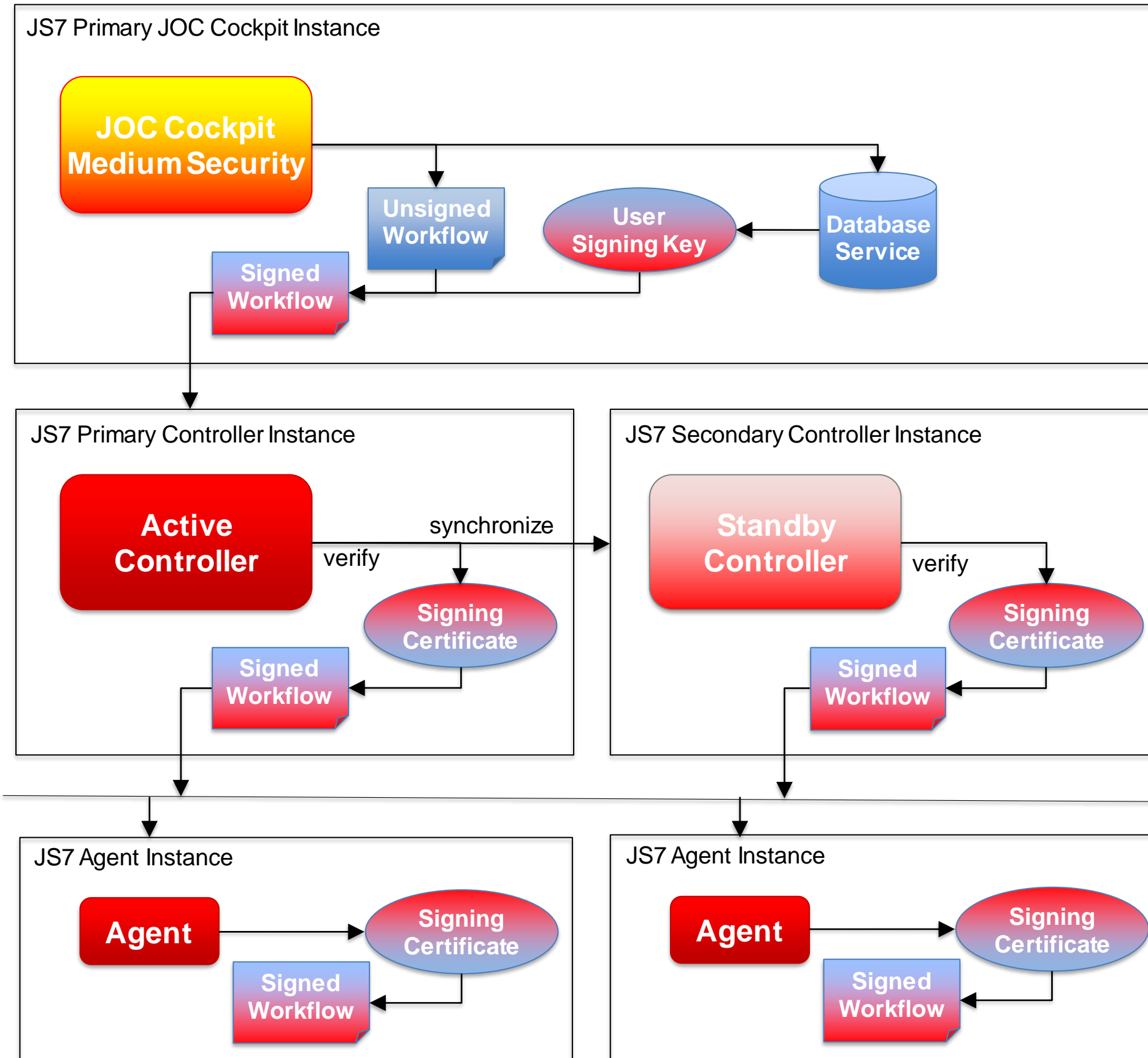
- Workflows are signed for deployment using a Signing Key from the database
- The signature is deployed with the workflow

Controller Signature Check

- Workflows are checked if they match the signature provided with certificates available to the Controller and otherwise are denied
- Similar check is performed by the Standby Controller that requires availability of the same certificates

Agent Signature Check

- Workflows are checked if they match the signature provided with certificates available to the Agent
- Workflows are not accepted in case of failed signature checks

**User based Signing**

- Workflows are signed individually per user for deployment with an individual Signing Key
- Deployment is a single click operation in the user interface

Security Level Medium

- JOC Cockpit offers built-in signing operations for workflows
- Each signed workflow is associated with a user
- The Signing Key might be copied or compromised as it is accessible from the database

External Signing

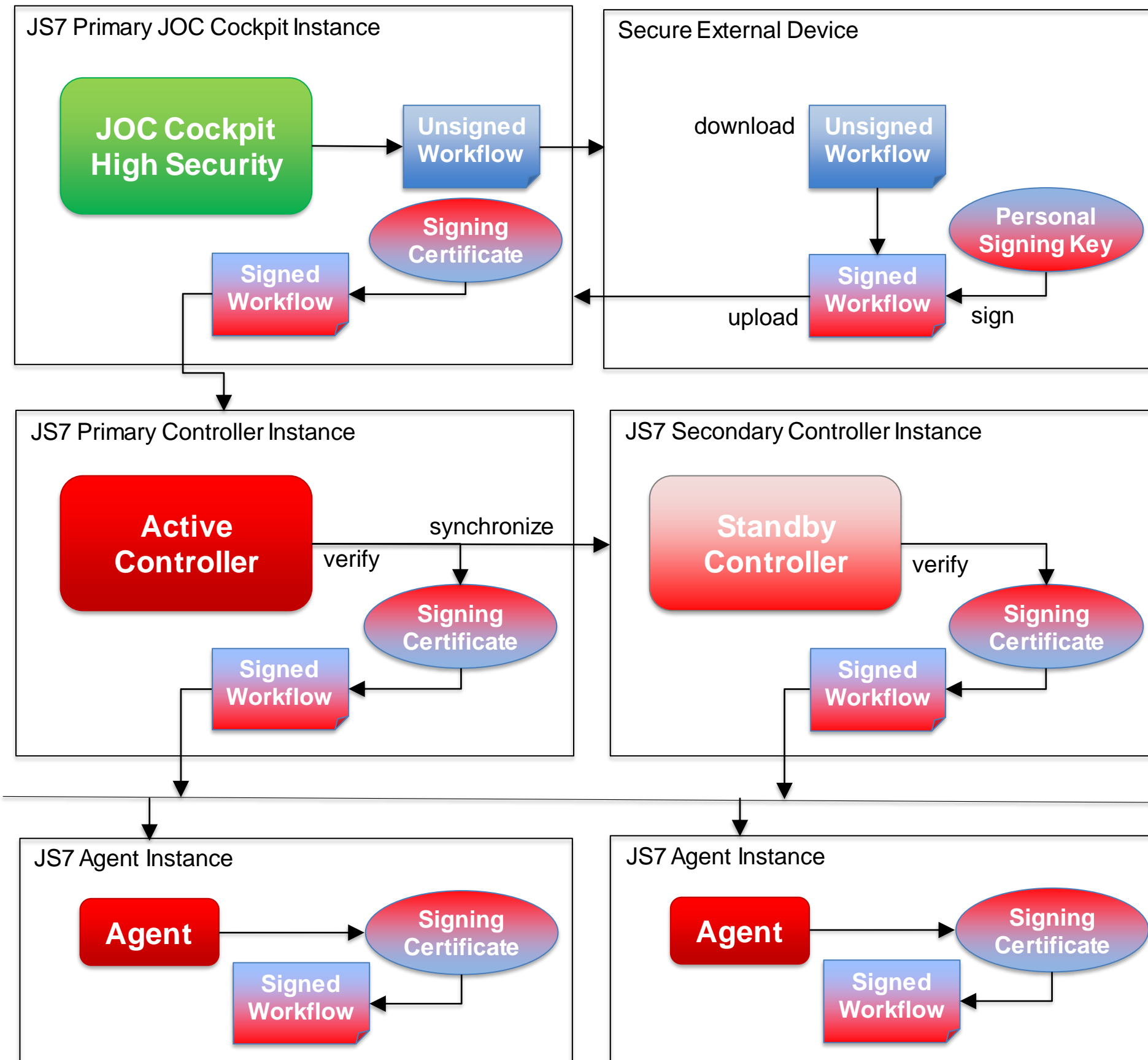
- Workflows are signed for deployment performing the signing from a secure external device.
- The signature is deployed with the workflow

Controller Signature Check

- Workflows are checked if they match the signature provided with certificates available to the Controller and otherwise are denied
- Similar check is performed by the Standby Controller that requires availability of the same certificates

Agent Signature Check

- Workflows are checked if they match the signature provided with certificates available to the Agent
- Workflows are not accepted in case of failed signature checks

**External Signing**

- Workflows are downloaded to a secure external computer, the user's Signing Key might be provided by portable media
- Signing is performed with any tool accepted by company standards, e.g. OpenSSL
- The workflows and resulting signature files are added to an archive file and are uploaded

Security Level High

- JOC Cockpit offers no built-in signing operations for workflows
- Workflows have to be signed externally to be deployable
- The Signing Key is used outside of JOC Cockpit
- A signed workflow is associated with a user (non-repudiability)

Secure Roll-out

External Signing

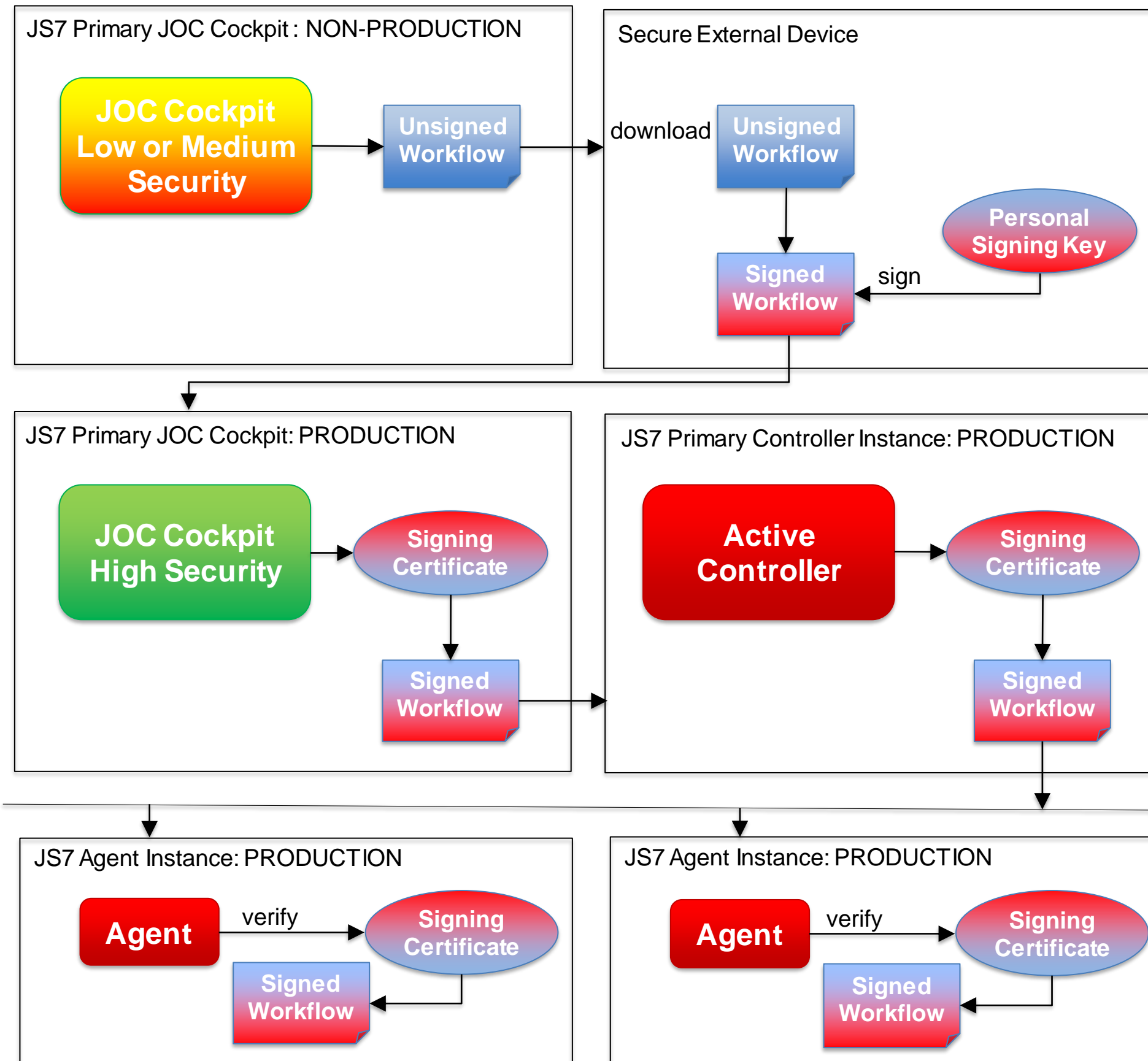
- Workflows are signed for deployment performing the signing from a secure, external device
- The signature is deployed with the workflow

Controller Signature Check

- Workflows are checked if they match the signature provided with certificates available to the Controller and otherwise are denied
- Similar check is performed by the Standby Controller that requires availability of the same certificates

Agent Signature Check

- Workflows are checked if they match the signature provided with certificates available to the Agent
- Workflows are not accepted in case of failed signature checks

**External Signing**

- Workflows are downloaded to a secure external computer, the user's Signing Key might be provided by portable media
- Signing is performed with any tool accepted by company standards, e.g. OpenSSL
- The workflows and resulting signature files are added to an archive file and are uploaded to a secure JOC Cockpit instance

Secure Roll-out

- A secure JOC Cockpit instance accepts signed workflows only
- There is no possibility to sign a workflow within a secure JOC Cockpit instance
- A secure JOC Cockpit instance will accept workflows only if their signatures match the signing certificate



Questions?
Comments?
Feedback?

Software- und
Organisations-
Service GmbH

Giesebrechtstr. 15
D-10629 Berlin

info@sos-berlin.com
<https://www.sos-berlin.com>