Product Sheet

Secure Connectivity Framework

Framework for connecting securely to Thredd systems

Secure access to Thredd's resources, using a common identity store.

Thredd's Secure Connectivity Framework is the combination of several components which enable secure access to Thredd's resources, using a common identity store. The main components are:

- **CloudEntity** – a Software as a Service (SaaS) capability which acts as the Identity Provider (IDP) for Thredd's interfaces (including the Thredd Certificate Authority and Thredd Portal), and as an OAuth OpenID Provider (OP) for the registration and management of customer applications, generation and validation of access tokens, and for the enforcement of access control policies.
    - **Thredd Certificate Authority or Thredd CA** – a SaaS capability for the creation and management of certificates:
        - Transport Certificates – for establishing secure connections between resources.
        - Signing Certificates – for the creation of signed messages, used for authentication of clients, and non-repudiation and authentication of notifications.
        - Encryption Certificates – for the encryption of payloads using an asymmetric encryption approach.
- **mTLS Termination** – on-premise infrastructure enabling the establishment of Trust Chains when clients present Thredd-issued Transport Certificates at the point of attempting to connect to protected resources.

"Zero Trust" Features

- **Standards based** – relies on OAuth, FAPI (financial grade API) and a range of associated RFCs. Where possible, Thredd's infrastructure is certified compliant with these standards.
- **Assume breach** – the security architecture design assumes that any network we host or are exposed to has been compromised.
- **Least privilege** – we grant users and systems the minimum level of access required for the performance of their function.
- **Authenticate everywhere** – we apply the same authentication requirements to external and internal users, whenever there is an attempt to access a protected resource.
- **Verify explicitly** – we continuously authenticate and authorise based on available data points, most notably user identity and constrained access.
- **Common approach** – we don't distinguish on the basis of internal / external status, but rather on the basis of resource sensitivity, and available verification data.
- **User-centric** – we aim to enhance the experience of those using our services, through self-service capability and complete documentation.

Approach to Network Security

- **Limited trusted network** – whilst our staff have access to a VPN, there are limited protected resources exposed on this by default. (We are in the process of deprecating client VPNs.)
- **Full tunnel for staff** – company devices can only access the internet through our company VPN, which includes a full tunnel. This enables the application of a range of protections at a network level, as well as ensuring traffic protection when connected to public access points.
- **PAM capability** – our infrastructure is not accessible directly, whether from the internet or over our company network. Connections can only be made through a hardened Privileged Access Manager (PAM), and access is granted on a per session basis.
- **Policy checks** – we authenticate devices attempting to connect to our resources, and apply a range of policy checks, both at the point of connection and on a continuous basis.
- **Micro-segmentation** – our security perimeters are broken up into a range of Virtual Private Clouds (VPCs) and zones. Traversing these requires authentication and authorisation, which is applied on a least privilege basis.
- **Offline production environment** – our Cardholder Data Environment (CDE) does not have access to the internet.

How It Works

Authentication

- **Non-shareable credentials** – we avoid using credentials which are shared (and could thus be intercepted and replayed) during the authentication process. Practically speaking, this means:
    - **Users** – we use a combination of possession and inherence based authentication factors for users' authentication. These are tied to passkeys which are frequently cycled.
    - **Client applications** – we have adopted *private_key_jwt* as the authentication method for applications, which relies on a signed (using a private key) *client_assertion* being sent to an authorisation server in place of static credentials.
    - **Storage** – after generation keys, secrets and credentials are stored in non-human accessible stores, and referenced as variables (as opposed to being stored in code).
- **Signature based validation** – to enable validation of message integrity and non-repudiation, we sign outbound notification messages. These can be verified using our public key.
- **Multi-factor** – we always rely on more than one credential for both user and service authentication.
- **Credential refresh** – credentials are refreshed regularly, using a risk-based approach. No single identity may have more than two credential sets live at any one time.

Authorisation

- **RBAC** – we enforce role-based access control (RBAC) across SaaS and proprietary applications, and rely on the hierarchy of roles, groups, scopes and claims. This applies both to users and client applications.
- **Constrained access tokens** – as part of our REST API service, we issue access tokens which are limited to certain scopes and claims. They cannot be replayed to resources which did not form part of the original token generation request.
- **Policy codification** – we use REGO to codify access control policy[1], meaning that it can be applied as code, and is easily transferable between Policy Decision Points (PDPs).
- **Regular privilege review** – as part of our preventative maintenance framework, we conduct regular reviews of both standard and privileged access. Where access is no longer required, it will be downgraded or removed.

Other

- **Encryption** – we take several approaches to data encryption:
    - **Transit** – all data in transit is encrypted at least at TLS1.2 and based on industry recommended best practice. Where sensitive data is being transferred between known parties, mutual TLS is employed.
    - **Storage** – all data in storage is encrypted by default, and (where appropriate) we employ PCI DSS specific requirements for encryption of cardholder data (CHD).
        - **Payload** – where API responses include cardholder data, we return encrypted payloads using a Public Key Cryptography-based, asymmetric encryption approach.
- **Monitoring** – for all connections and traffic, we maintain a high degree of visibility, and pipe logs to our Security Information and Event Management (SIEM) capability. This is overseen by a 24/7 Security Operations Center (SOC), with continuous development of use cases, and refinement of alerts.
- **Pre-emptive strike** – as part of our endpoint management and identity access management capabilities, we have configured systems to take a range of actions automatically, from step-up authentication to blocking access and isolating devices.
- **Acting on threat intelligence** – we rely on third-party advisers and market intelligence data feeds to update our approaches to the evolving threat landscape.

---

[1] For more information on Rego, see the Openpolicyagent.org.

Standards Alignment

NCSC ZTA Alignment

The National Cyber Security Centre (NCSC) Zero Trust Architecture (ZTA) guidance has been followed across Thredd's implementation.

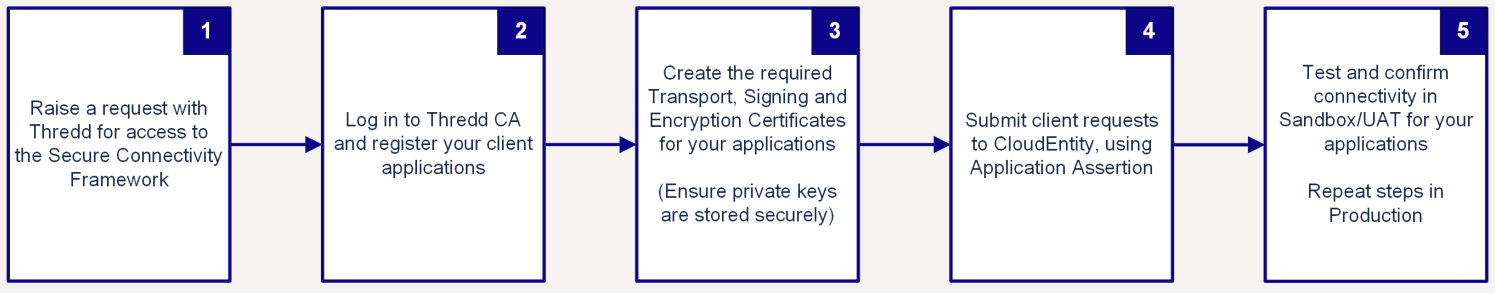
Cybersecurity and Infrastructure Security Agency (CISA) Maturity Model

Thredd has aligned to an 'optimal' standard, for implementation across its Zero Trust approach / security architecture.

	Identity	Devices	Networks	Applications and Workloads	Data
Optimal	• Continuous validation and risk analysis • Enterprise-wide identity integration • Tailored, as-needed automated access	• Continuous physical and virtual asset analysis including automated supply chain risk management and integrated threat protections • Resource access depends on real-time device risk analytics	• Distributed micro-perimeters with just-in-time and just-enough access controls and proportionate resilience • Configurations evolve to meet application profile needs • Integrates best practices for cryptographic agility	• Applications available over public networks with continuously authorized access • Protections against sophisticated attacks in all workflows • Immutable workloads with security testing integrated throughout lifecycle	• Continuous data inventorying • Automated data categorization and labeling enterprise-wide • Optimized data availability • DLP exfil blocking • Dynamic access controls • Encrypts data in use
	Visibility and Analytics		*Automation and Orchestration*		*Governance*
Advanced	• Phishing-resistant MFA • Consolidation and secure integration of identity stores • Automated identity risk assessments • Need/session-based access	• Most physical and virtual assets are tracked • Enforced compliance implemented with integrated threat protections • Initial resource access depends on device posture	• Expanded isolation and resilience mechanisms • Configurations adapt based on automated risk-aware application profile assessments • Encrypts applicable network traffic and manages issuance and rotation of keys	• Most mission critical applications available over public networks to authorized users • Protections integrated in all application workflows with context-based access controls • Coordinated teams for development, security, and operations	• Automated data inventory with tracking • Consistent, tiered, targeted categorization and labeling • Redundant, highly available data stores • Static DLP • Automated context-based access • Encrypts data at rest
	Visibility and Analytics		*Automation and Orchestration*		*Governance*
Initial	• MFA with passwords • Self-managed and hosted identity stores • Manual identity risk assessments • Access expires with automated review	• All physical assets tracked • Limited device-based access control and compliance enforcement • Some protections delivered via automation	• Initial isolation of critical workloads • Network capabilities manage availability demands for more applications • Dynamic configurations for some portions of the network • Encrypt more traffic and formalize key management policies	• Some mission critical workflows have integrated protections and are accessible over public networks to authorized users • Formal code deployment mechanisms through CI/CD pipelines • Static and dynamic security testing prior to deployment	• Limited automation to inventory data and control access • Begin to implement a strategy for data categorization • Some highly available data stores • Encrypts data in transit • Initial centralized key management policies
	Visibility and Analytics		*Automation and Orchestration*		*Governance*
Traditional	• Passwords or MFA • On-premises identity stores • Limited identity risk assessments • Permanent access with periodic review	• Manually tracking device inventory • Limited compliance visibility • No device criteria for resource access • Manual deployment of threat protections to some devices	• Large perimeter/macro-segmentation • Limited resilience and manually managed rulesets and configurations • Minimal traffic encryption with ad hoc key management	• Mission critical applications accessible via private networks • Protections have minimal workflow integration • Ad hoc development, testing, and production environments	• Manually inventory and categorize data • On-prem data stores • Static access controls • Minimal encryption of data at rest and in transit with ad hoc key management

Setup At-a-Glance

Below is a high-level view of the setup steps for access to the Secure Connectivity Framework.



Setup Options

The table below provides details of the certificates for generating and storing for connecting to Thredd applications, and where CloudEntity is required.

Thredd Application	Transport Certificate	Signing Certificate	CloudEntity
Thredd Portal	x (pre-installed)	x	√
REST API	√	√	x
SOAP API	√	x	x
External Host Interface (EHI)	x (provided by Thredd) Root and Issuing certificates required	x	x

FAQs

Q. What is the Secure Connectivity Framework?

The Secure Connectivity Framework is the combination of several components which enable secure access to Thredd's resources, using a common identity store.

Q. Who is Thredd's Certificate Authority (CA)?

Thredd's Certificate Authority is its own Certificate Authority. Using the PKI, Thredd's customers can acquire Signing Certificates (used for *private_key_jwt* authentication) and Transport Certificates for mTLS connections.

Q. What is the role of CloudEntity?

CloudEntity is used for several purposes:

- A central hub Identity Provider / Authorisation Server for Users to authenticate with, including federated authentication using the Single Sign On (SSO) capabilities of Thredd's customers.
- An Identity Pool (multiple identity pools, one per organisation) with Users and their Roles all managed in one place.
- An Authorisation Server for Server to Server communication (for Confidential Clients to gain Access Tokens)
- A Policy Decision Point (PDP) to Allow / Deny access to protected Thredd Resources (Core REST APIs) based on policies which check attributes of incoming REST requests including Access Token Claims, mTLS Certificates, and User Roles.

Q. Can I still use or request a VPN setup?

No. You must connect securely to Thredd using the Secure Connectivity Framework.

Q. Where can I find out more?

For more information, refer to the Connecting to Thredd Guide or contact your Account Manager.

Contact Us

Please contact us if you have queries relating to this document. Our contact details are provided below.

Thredd UK Ltd.

Support Email: occ@thredd.com

Telephone: +44 (0) 203 740 9682

Our Head Office

Kingsbourne House

229-231 High Holborn

London

WC1V 7DA

Technical Publications

If you want to contact our technical publications team directly, for queries or feedback related to this guide, you can email us at: docs@thredd.com.