

## nevisProxy – Dynamic Session Management Engine

### Benefits

The Dynamic Session Management Engine of nevisProxy deals with session handling. It tackles all kinds of issues regarding the management of sessions. For instance, the engine:

- Prevents unnecessary updates or reloads of (website) pictures, style sheets or JavaScript code that occur in combination with a remote session database, and thus considerably **improves performance** of your system.
- Prevents client-server connections from staying alive and open even if there is no user activity for a long time, and thus considerably **increases system security**.
- Offers considerably **more flexibility in handling sessions**, because it
  - Allows for the parallel existence of secured sessions as well as connections over session free paths.
  - Supports mixtures of remote and local session storage with different storage requirements (e.g., for multi-tenant solutions).

Additionally, the engine prevents unintended session overwrites and data loss when multiple requests within the same session run in parallel.

---

### What is it about?

A session is the period of time between the login and the logout of a client to an application protected by NEVIS. Within NEVIS, it is nevisProxy that is in charge of session handling and management. Session management includes binding the session to the client (e.g., with a session cookie); storing session information such as client identity, cookie, security roles and routing information into a local or remote cache; handling timeout situations, and terminating the session.

nevisProxy decides on the strategy of handling the session for a request based on the identity and login information of the client as well as the requested URL-address.

Because each type of request and application has its own requirements regarding session handling, there is a need for a diversified session management. The Dynamic Session Management Engine satisfies this need: It allows you to define different session management

rules for different applications and requests, and supports the use of local and remote session storage at the same time. Due to this high flexibility in handling sessions, you will reach a higher system security and better performance.

nevisProxy also offers a second session handling method, the Container Session Handling. The difference between this session handling method and the Dynamic Session Management Engine lies in the architectural concept as well as the location of the session creation and handling within the processing of a client request.

Because of its different architecture, the Container Session Handling facilitates just one strategy of session handling with a limited set of rules and one session storage method. Compared to this, the Dynamic Session Management Engine provides a much more powerful and adaptable way of managing sessions.

---

### Main features

- Powerful and sophisticated session management engine.
- High flexibility in handling sessions.
- High adaptability of the session management to different kinds of requests.
- Allows for adjustments to special application requirements regarding sessions.
- Supports setups with multiple session handling variations.

Architecture

Figure 1 shows the architecture of the Dynamic Session Management Engine:

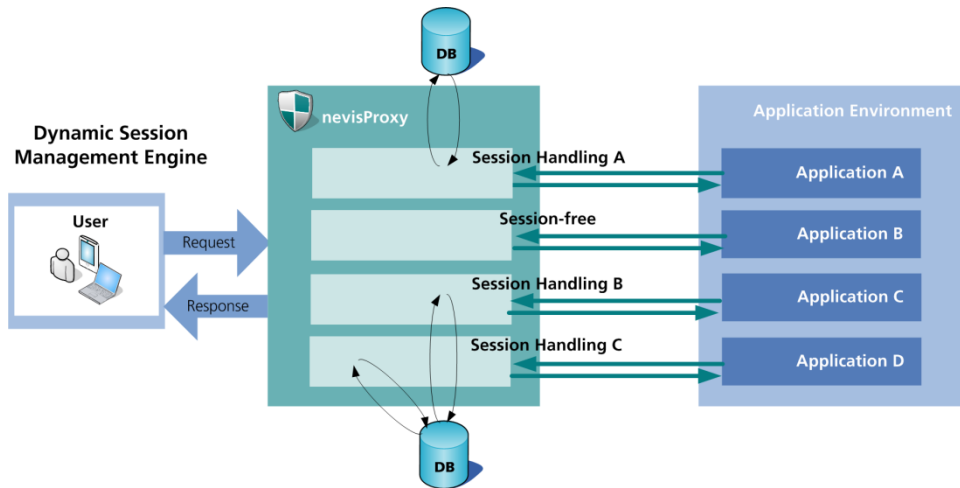


Figure 1 Architecture overview