

# Diameter Stack Design

# Table of Contents

Diameter Stack Extensibility .....	1
Diameter Stack Model .....	1
Application Session Factories .....	2
Session Replication .....	3

# Diameter Stack Extensibility

Diameter Stack has been designed to be extensible. In order to achieve that, two set of APIs are defined by the stack - one that defines basic contracts between the user application and the stack, and one that defines contracts allowing the instance to inject custom objects into the stack to perform certain tasks (for example, `SessionFactory`). `ISessionFactory` declares additional methods that allow the developer to declare custom behaviour (for example, custom application sessions). Please refer to [\[jdiameter\\_source\\_overview\\_session\\_factory\]](#) for more detailed information.

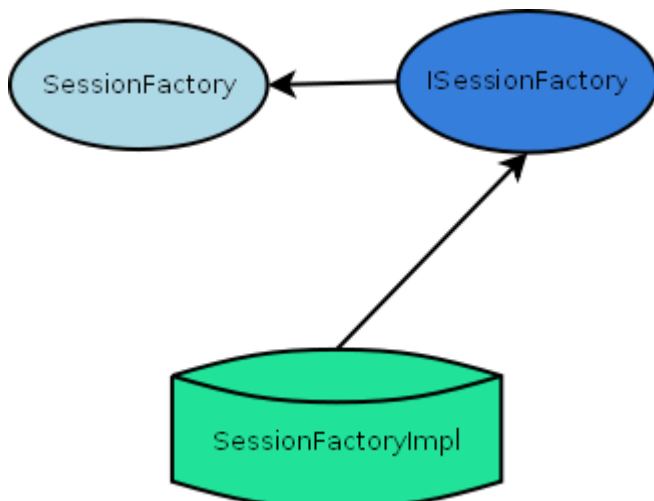


Figure 1. Diameter Stack Extensibility Visualization

The general pattern for interface declaration is:

- The interface `ComponentInterface` declares the minimal set of methods for a component to perform its task.
- The interface `IComponentInterface` provides additional behavior methods. Please refer to the java doc for a list of interfaces and descriptions of method contracts.

## Diameter Stack Model

Diameter Stack performs the following tasks:

- Manages connections to remote peers.
- Manages session objects.
- Routes messages on behalf of sessions.
- Receives and delivers messages to assigned listeners (usually a session object).

Sessions use the stack and the services it provides to communicate with remote peers. The application is the only place that holds references to sessions. It can be seen as follows:

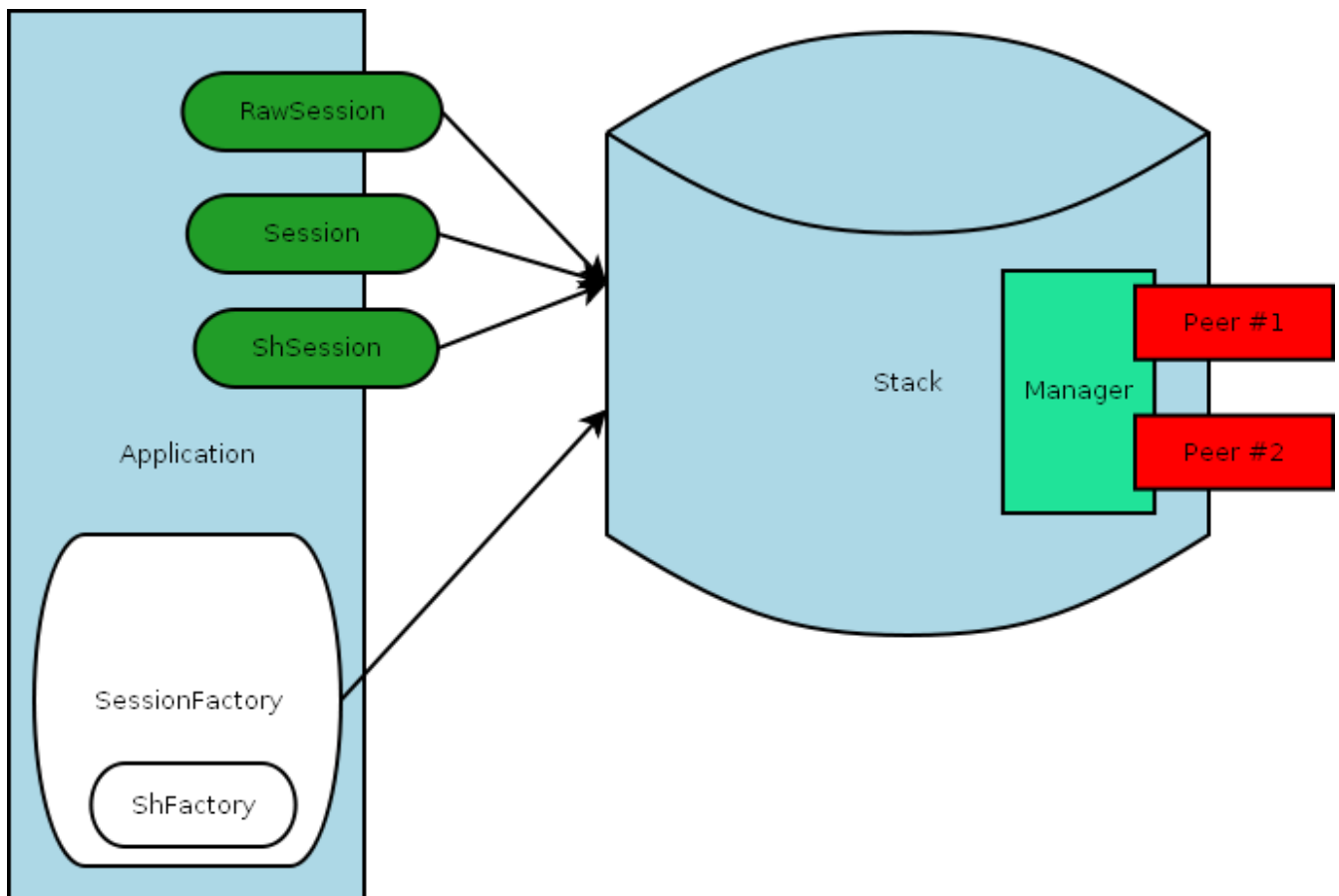


Figure 2. Diameter Application and Stack Model

## Application Session Factories

Application session factories perform two tasks:

- server stack as factory for sessions.
- server session objects as holders for session related resources, like state change listener, event listeners and context.

The figure below shows the relationship between Application Session Factories and User Applications:

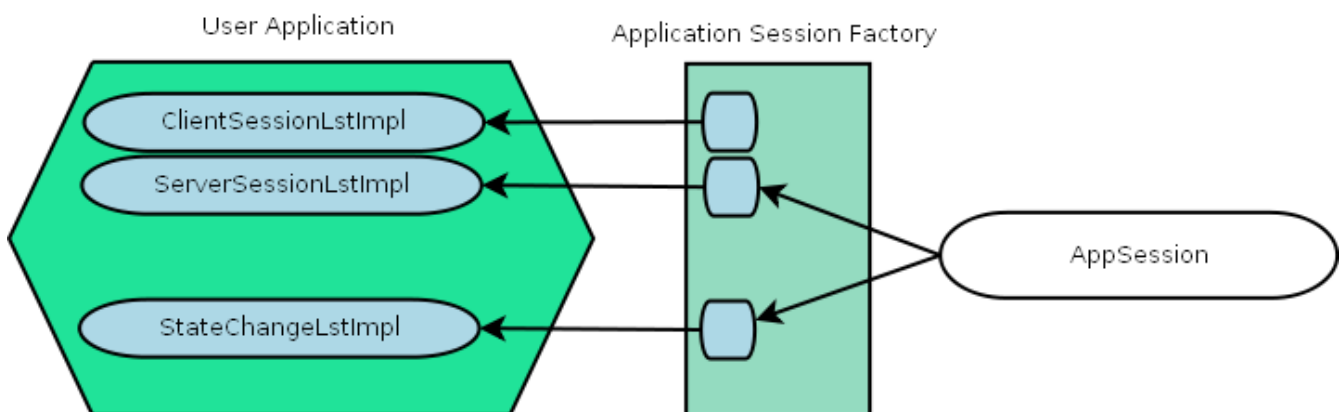


Figure 3. Application Session Factory and User Application



Session context is a callback interface defined by some sessions.

# Session Replication

Diameter Stack supports replication of session data and state. Clustered stack instances can perform operations on session regardless of physical location. Imagine the logically clustered stack as follows:

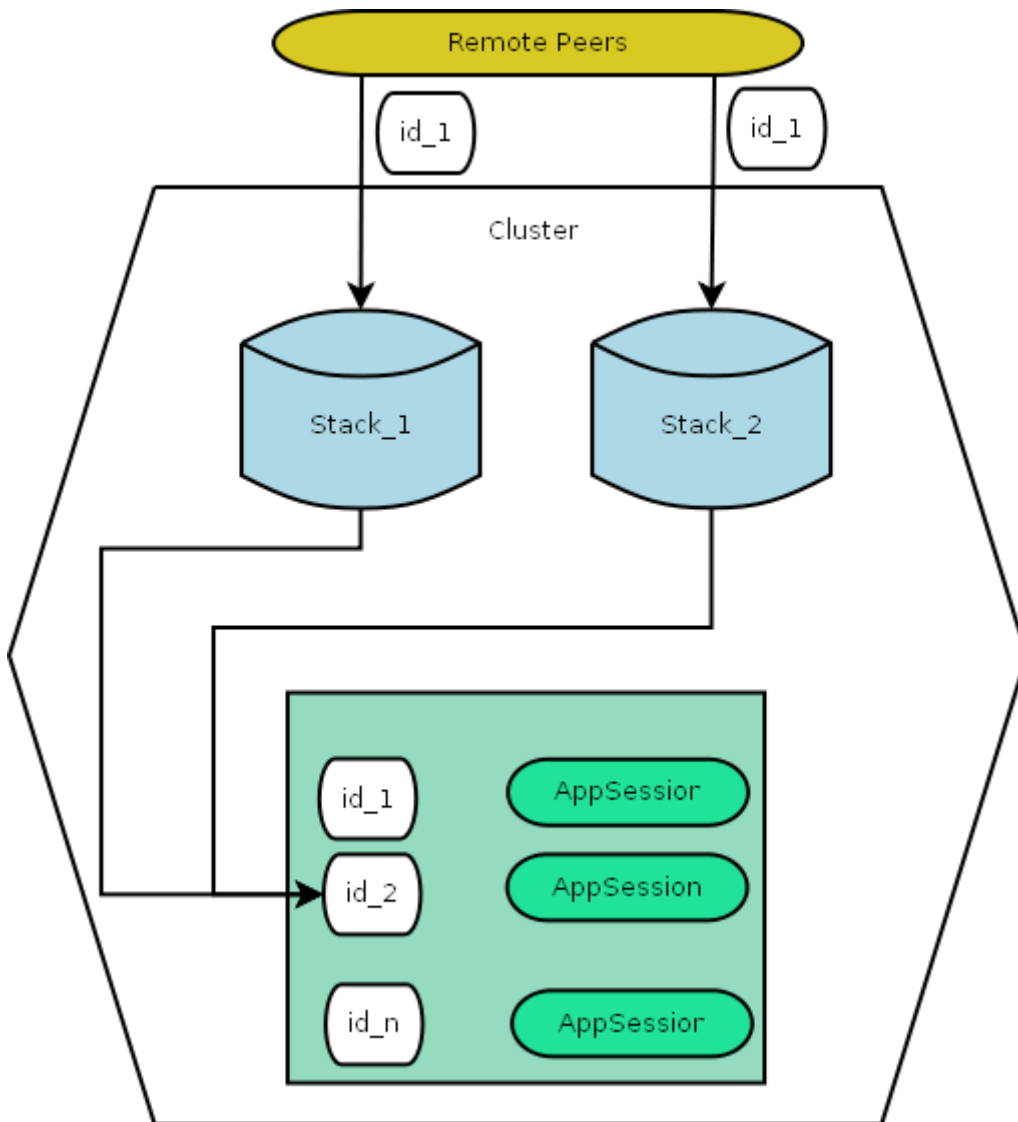


Figure 4. Diameter Cluster

Stack only replicates non simple sessions. This is because simple sessions do not hold state and can be simply recreated by the application. Simple sessions include:

- RawSessions
- Sessions

Diameter Cluster replicates the full state of sessions. However, it does not replicate resources that are entirely local to the stack instance, like session listeners. Local resource references are recreated once the session is being prepared to be used in the stack instance. Listeners (state and events) are fetched from the respective session factory instance. See [Application Session Factories](#) for more details.