neo system architectures

Installation manual

for system providers

10/18/2021

Product line *neo*, version 6.x

The described functions can be used with the following ASC products:

EVOIP*neo*

EVOLUTION*neo* / XXL / eco

EVO*flex* (country-specific)

Please note that you can always find the most up-to-date technical documentation and product updates in the partner area on our website at http://www.asctechnologies.com.

ASC Technologies AG - Seibelstr. 2-4 - 63768 Hösbach - Germany

Contents

1 General information

In the context of this document ASC represents ASC Technologies AG, its subsidiaries, branch offices, and distributors. An up-to-date overview of the aforementioned entities can be found at https://www.asctechnologies.com

ASC assumes no guarantee for the actuality, correctness, integrity or quality of the information provided in the manuals.

ASC regularly checks the content of the released manuals for consistency with the described hardware and software. Nevertheless, deviations cannot be excluded. Necessary revisions are included in subsequent editions.

Some aspects of the ASC technology are described in general terms to protect the ownership and the confidential information or trade secrets of ASC.

Representative, if not defined otherwise, is the technical status at the time of the delivery of the software, the devices and the manuals of ASC. Technical changes without specified announcements are reserved. Previous manuals lose their validity.

The general conditions of sales and delivery of ASC in their latest version apply.

2 Introduction

The recording system creates recordings of conversations which are conducted via dedicated communication platforms.

The individual components of the system can either be installed on one single server or on several distributed servers. This results in multiple system architectures, see chapter "Possible system architectures", p. 22.

Within the different system architectures, different recording architectures can be used, see chapter "Color scheme for system components", p. 9.

The deployed servers can be situated in different locations. The location where the conversations are recorded and the location where the recording system is controlled do not have to be the same. Recordings can be searched for and replayed locally as well as via the network.

For a safe, uninterrupted recording and optimized distributed load sharing, the system provides several levels to set up individual components redundantly, see chapter "Redundancy options", p. 21.

3 General mode of operation

The **recording system** basically consists of the following components:

- Web-based user interface with access to the different *neo* applications
- Enterprise Core with the application server (app server)
- Recording architecture with recording components
- Database

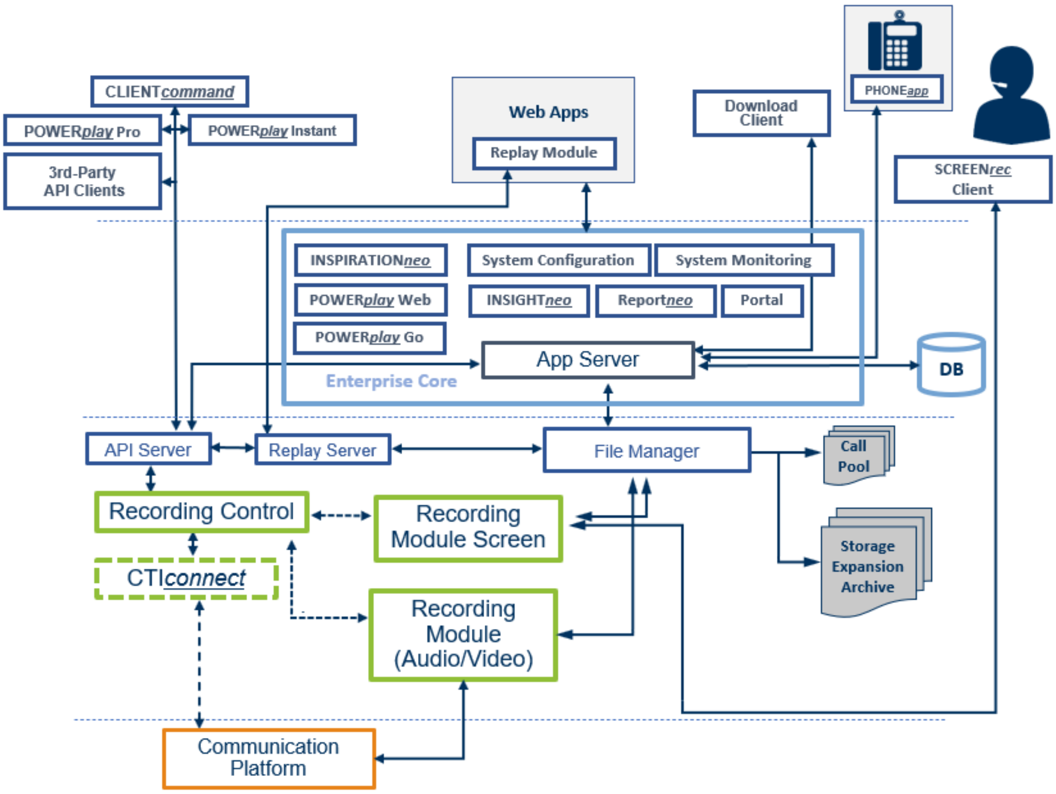

Fig. 1: Basic structure of the recording system

The following components may be installed on different servers:

- App server
- Recording components and API server
- Database

This results in several different system architectures.

For information about which system architectures are possible in general, see chapter "Possible system architectures", p. 22.

The data stream of the recording process usually follows the pattern described below:

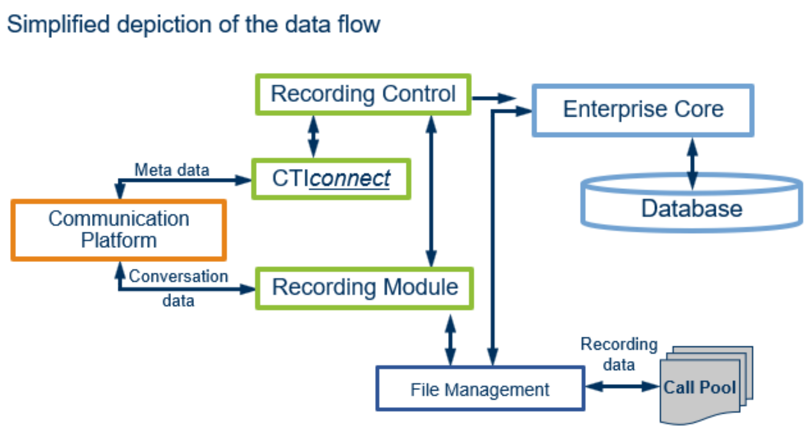

Fig. 2: Data stream in the recording process

The data stream of the recording depends on the integration type; therefore, a separate integration has to be created in the recording system for each integration type.

Information about the configuration of the different integration types can be found in the respective integration-specific administration manual.

Every integration uses a **recording architecture** for the recording. A recording architecture always includes the following recording components:

- Recording Control

  This service controls the recording according to the recording plan.

- CTI*connect* (optional)

  This service receives additional data about the recordings from the communication platform.

- Recording Module

  This service creates the recording data. The server that this service has been installed on is called a recording server.

The setup of a recording architecture defines the way in which the recording components interact. Some architecture types offer the possibility to install recording components redundantly.

In addition in some recording architectures, the individual recording components may be activated on different servers.

The different supported architecture types have been stored in the system and serve as the basis for defining the individual recording architectures of the system providers.

For information about which recording architectures have been stored in the system, see chapter "System architecture types", p. 10.

For information about the configuration of the individual recording architectures refer to the installation manual *Configuration servers and recording architectures*.

4 Color scheme for system components

The following color scheme is used to illustrate system components:

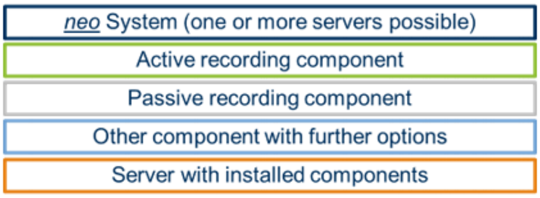

Fig. 3: Color scheme system components

The arrows in the images depict the communication channels between the components.

5 System architecture types

The *neo* system supports the following system architecture types:

5.1 Basic system architecture types

5.1.1 All-in-one Basic

With a system architecture of this type, all recording components are located on one single server. Additional components such as the Enterprise Core or the database may be installed on this server as well. There are no redundant recording components.

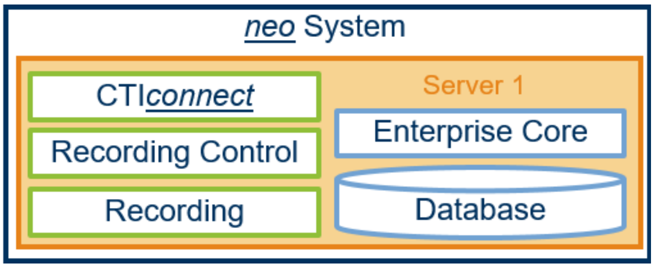

Fig. 4: System architecture with All-in-one Basic recording architecture with one server

In an All-in-one Basic recording architecture, all recording components have been installed on one server. To increase the recording capacity, the Enterprise Core and the database can be installed on a second server. A redundancy is not possible in this constellation; however, the full capacity of the first server can be used for the recording functionalities.

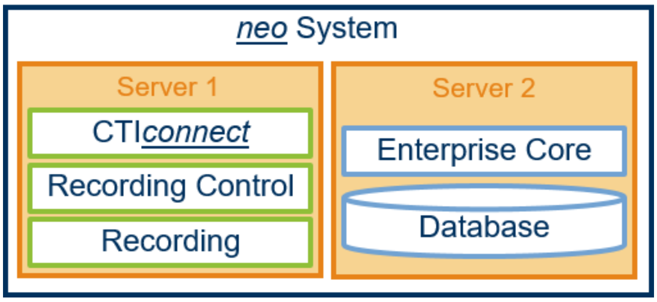

Fig. 5: System architecture with 2 servers with All-in-one Basic recording architecture

5.1.2 Multi-Server Recording

A system architecture of this type allows distributed load sharing across several recording servers. The pool of recording servers may comprise any number of recording servers. Other recording components (Recording Control Service and CTI*connect*) can also be set up redundantly.

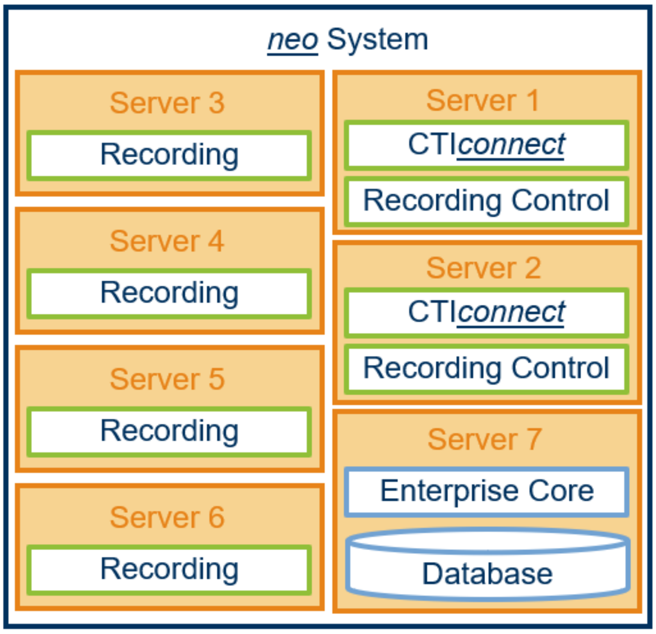

Fig. 6: System architecture with 7 servers, Multi-Server Parallel Recording recording architecture and separate Enterprise Core with database

The recording components on different servers can record different recording lines or can be configured for parallel recording as redundancy.

5.2 Architecture types for failover concepts

A failover recording architecture serves the purpose of providing you with a functioning recording system as soon as possible after a recording component has failed to minimize the amount of lost recordings. To this end, two recording trunks are installed only one of which is active at the same time, though. One recording trunk is configured as primary recording trunk. If a recording component of the primary recording trunk fails, the standby recording trunk automatically takes over the recording. The application server controls the switch from the primary to the standby recording trunk.

ATTENTION!

In failover architectures in which several integrations are active, all integrations of this recording architecture are switched to the other system in case of an error.

The import function works only on servers on which a Recording Control Service is running.

An import does not take place when switching to a server without a Recording Control Service in case of an error.

An import does not take place when switching to a server with a Recording Control Service but without a configured import function in case of an error.

If the standby server which has taken over the active role fails, then the system does not switch to the primary server automatically even if it should be operative again.

If you would like to switch back to the original primary server again as soon as it is operative, you have to configure this option manually, see Standby management for failover architectures.

If you do not want to switch back, you can run the active standby server as primary server. To ensure that the system switches back automatically from the standby server to the original primary server, the option *Activate standby failover* must be active in the recording architecture, see Create recording architecture.

ATTENTION!

If errors occur during failover operation on the activated standby recording components, recordings are inevitably lost.

5.2.1 All-in-one Failover

An All-in-one Failover architecture consists of two servers.

On Server 1, the Enterprise Core and the database as well as the recording components are installed.

On Server 2, only the recording components are installed and activated as primary components.

If one of the primary recording components on Server 2 fails, the Enterprise Core activates the standby recording components on Server 1 so that recording can continue.

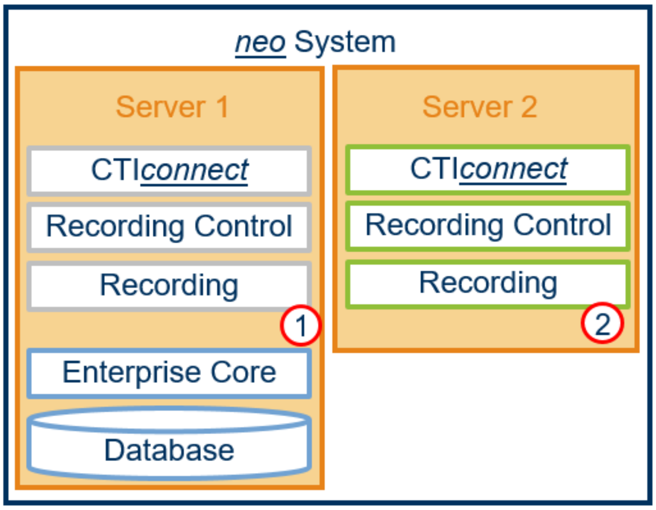

Fig. 7: System architecture with All-in-one Failover recording architecture

To be able to initiate the failover mechanism, a signal must be sent from the Enterprise Core to the modules which are supposed to be started. The signal can only be sent if there is a connection between the servers or if the services are running locally on the server of the Enterprise Core. Therefore, the primary recording modules have to run on the separate Server 2. The recording modules on the Enterprise Core Server 1 have to be configured as standby so that the services on Server 1 can be started in case the primary recording modules on Server 2 fail.

Configure alarm messages so that you are notified about failover operations and will be able to take respective measures.

After a failover case, you must switch back to server 2 manually.

For basic information about the Notifications module refer to the administration manual for tenants *Notifications module*.

5.2.2 Multi-Server Failover

In a failover architecture of this type, the recording components of the two recording trunks are distributed on several servers.

The recording components Recording Control and CTI*connect* have been installed twice and thus offer a simple redundancy.

To distribute the load, a pool of recording servers can be created which can contain any number of recording servers. The pool of recording servers can be set up once or twice or with redundant components.

The architecture type *Multi-Server Failover* allows realizing the following architecture scenarios:

Multi-server failover with redundant recording control and one recording server pool

There is 1 pool of recording servers. This pool of recording servers can be controlled by both Recording Control Services.

If the primary Recording Control Service fails, the Recording Control Service of the standby recording trunk becomes active. It takes over the control of the pool of recording servers.

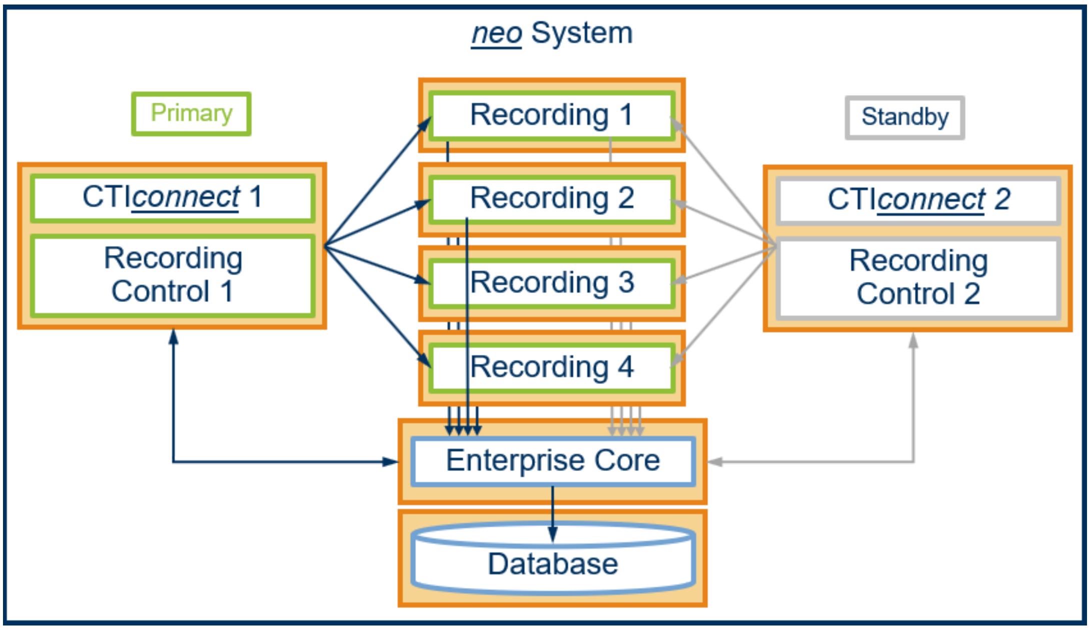

Fig. 8: System architecture with Multi-Server Failover recording architecture with a pool of recording servers

Multi-Server Failover with full recording server redundancy

There are 2 pools of recording servers. Each recording server has an assigned standby recording server which takes over the function of the primary recording server if the latter fails. In the following exemplary figure, the recording server with *Recording Module 1b* is the standby server for the recording server with *Recording Module 1a* while the recording server with *Recording Module 2b* is the standby server for the recording server with *Recording Module 2a*.

If the primary Recording Control Service fails, the Recording Control Service of the standby recording trunk becomes active. It takes over the control of the pool of recording servers. If a recording server within the pool of recording servers fails, the unambiguously defined standby recording server takes over its function regardless of the Recording Control Service which is currently active.

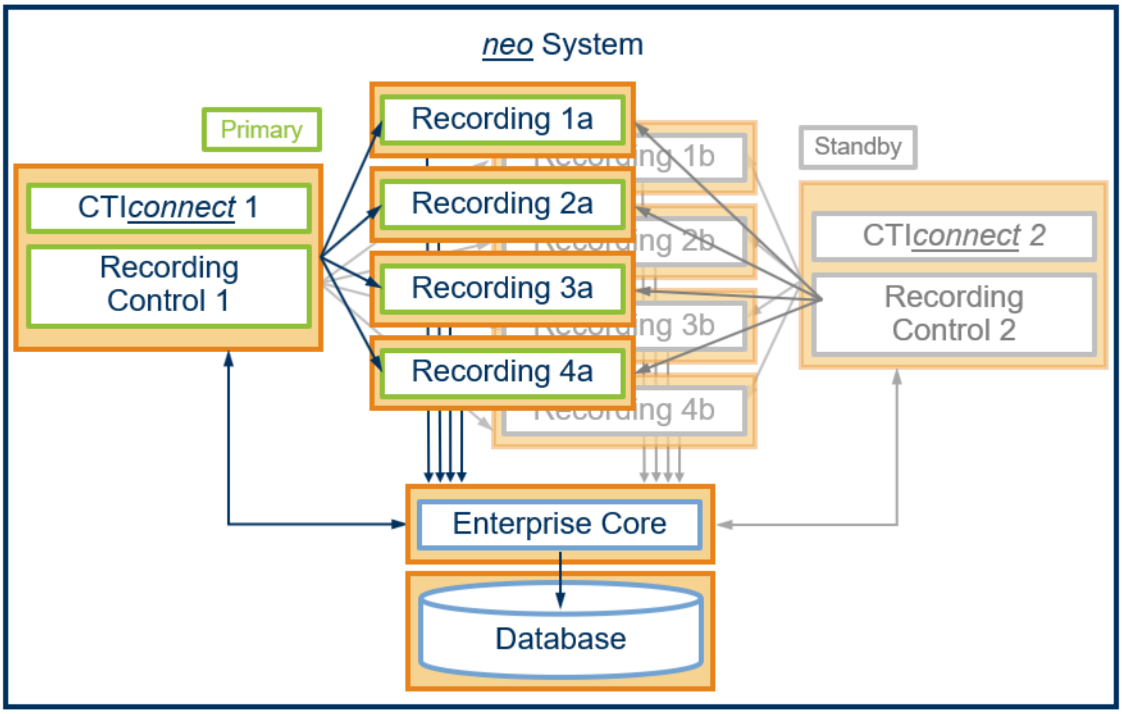

Fig. 9: System architecture with Multi-Server Failover recording architecture with redundancy options

5.3 Architecture types for parallel recording

Parallel recording serves to avoid loss of recordings when a recording component fails. For this purpose, 2 recording lines are configured which are active simultaneously so that recordings are created twice. If one recording line fails, a recording is created by the other recording line.

There are different configuration possibilities for parallel recording architectures:

- *Parallel recording without synchronization*

  A recording server cannot take over recording control from another recording server. Double recording can be deleted from the system by means of the function “Delete duplicates”.

- *Parallel recording with synchronization*

  The Recording Control modules of the two recording servers are synchronized. If one recording server fails, the other recording server can take over recording control.

  In this case, only one conversation is visible in the players but 2 audio files have been saved in the background. The option “Delete duplicates” is not possible in this recording variant, therefore twice as much storage capacity is required.

5.3.1 All-in-one Parallel Recording

The smallest setup of this architecture consists of a minimum of two servers which each contain all recording components. Recording takes place on both servers in parallel. In case of a failure, there is no need to switch to another architecture and recording can be guaranteed without interruption. Enterprise Core and database may be installed on one of these servers as well.

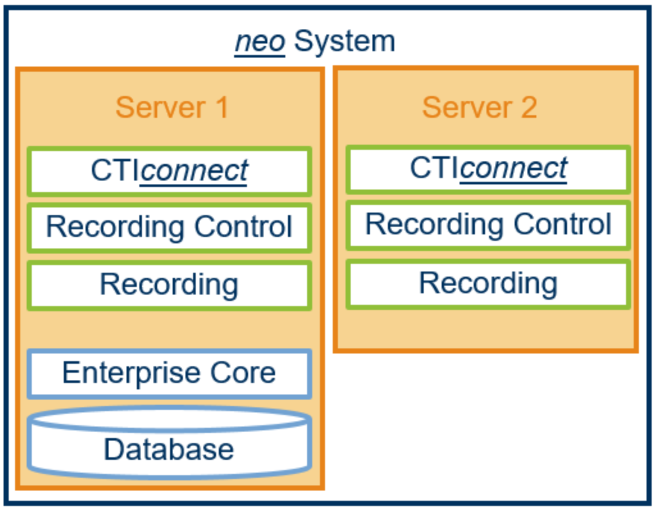

Fig. 10: System architecture with All-in-one Parallel Recording recording architecture

However, ASC recommends to install the Enterprise Core along with the database on a third server. Neither of them is redundant but can be expanded accordingly.

All-in-one Parallel Recording with 3 servers

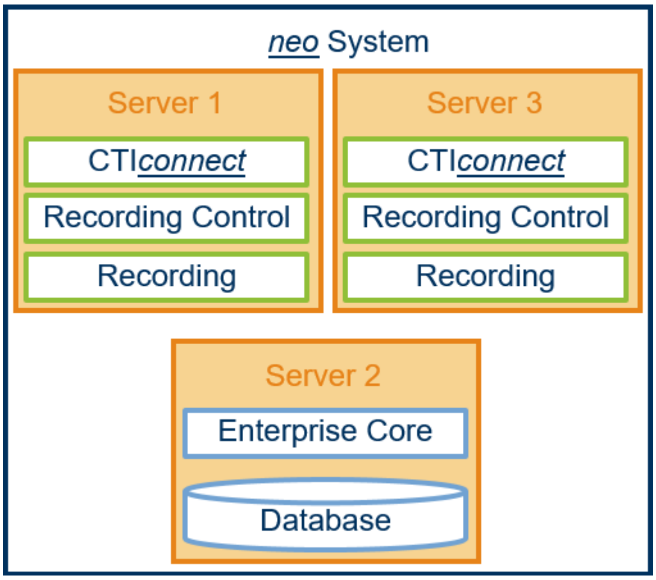

Fig. 11: System architecture with All-in-one Parallel Recording recording architecture with 3 servers

5.3.2 Multi-Server Parallel Recording

In an architecture of this type, the recording components of the two recording lines are distributed among several servers. Each recording line has its own pool of recording servers.

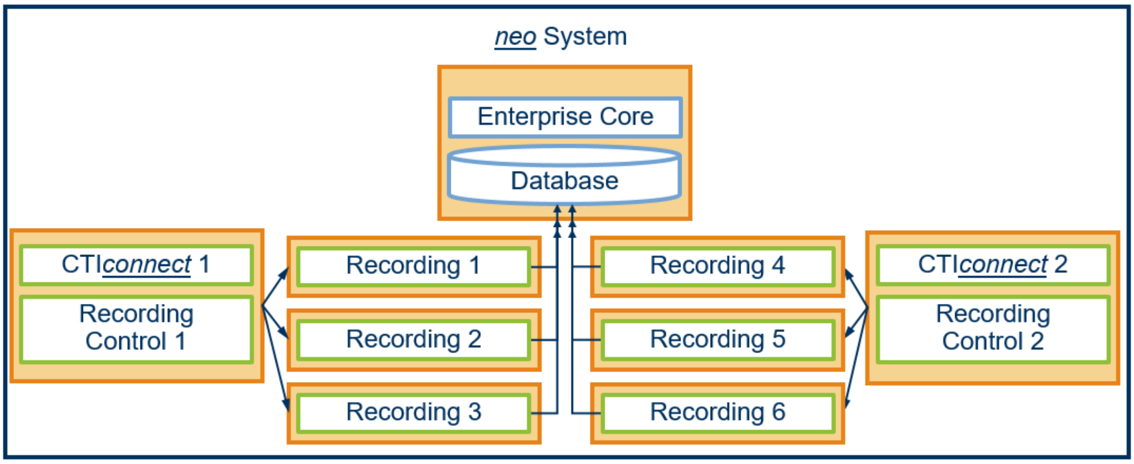

Fig. 12: System architecture with Multi-Server Parallel Recording recording architecture

5.4 Architecture types for import

The architecture type *Import Only* allows configuring a simple recording architecture especially for import. The usage of the architecture type is useful if the system is not used for recording and the recording data will exclusively be imported.

This recording architecture type cannot be used for recording as it contains neither recording servers nor Recording Module. This architecture type exclusively consists of 1 recording component *RecordingControl*.

> Theoretically, an import works with any other architecture type, too. As a result, you may use a recording architecture for import which is already used for an integration.

6 Synchronization options

The are 2 different types of synchronization:

- Synchronization of the Recording Control Service for recording control
- Synchronization of the system storage to compare recording data

6.1 Synchronization of recording control

Recording Control Services

For parallel recording servers installed in the same system architecture, you can configure synchronization of recording control.

ATTENTION!

Before the configuration, contact your ASC support to ensure that this function is suitable for your recording solution and to avoid a possible loss of recordings!

For information about which recording solutions support this function refer to the file *neo* Integration Overview.

If recording is supposed to be controlled manually by means of applications such as CLIENT*command*, PHONE*app*, SCREEN*rec* scan Editor, or by external control, synchronization of the Recording Control Services of the parallel recording servers must be created.

Initially, the 1st Recording Control Service is responsible for recording control. The Recording Control Service ensures that the conversations of both recording servers are recorded.

If the 1st Recording Control Service fails, the 2nd Recording Control Service takes over recording control for both recording servers each of which then records the conversations.

Synchronization of recording control is configured in the Recording Architectures module. In parallel recording architectures, the check box *Synchronize recording control* appears in the tab *Details*.

1. Activate the check box *Synchronize recording control* so that the Recording Control Services can be synchronized and only one service controls recording for the two recording servers.

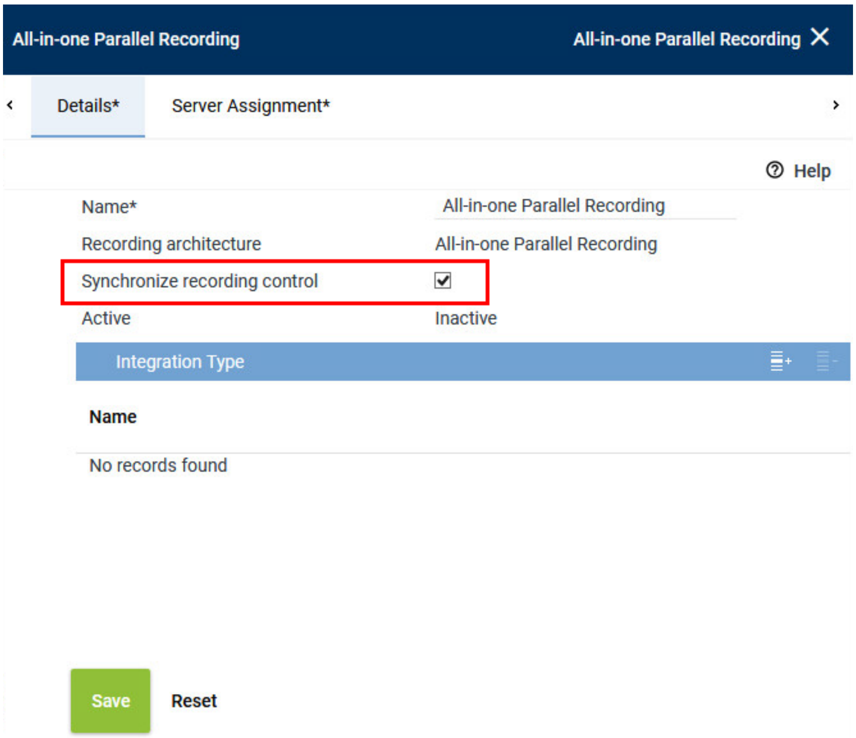

Fig. 13: Synchronize recording control

2. To save the settings, click on the button *Save*.
   To discard the settings, click on the button *Reset*.

Synchronization of recording control brings stricter timeouts between the components. Observe the increased hardware and network requirements. Latency must be < 100 ms.

If you activate or deactivate this synchronization option subsequently, you must repeat the following configuration steps for the changes to take effect:

1. Select the required state of recording control:
   ☑ = *Recording control is synchronized*
   ☐ = *Recording control is not synchronized*
2. Deactivate the integration.
3. Deactivate the recording architecture.
4. Ensure that the following services have been stopped:
   - *ASC RecordingControl*
   - *ASC RecordingModule*
   - *ASC CTIconnect(integration name)*
5. Activate the recording architecture.

WARNING! In this status, all services have received the updated configuration but states may be conflicting.

Therefore, repeat the following steps:

6. Deactivate the recording architecture again.
7. Ensure that the services have been stopped.
8. Activate the recording architecture again.
9. Activate the integration.

⇨ The changes are now active.

6.2 Synchronization of system storage

In recording architectures with 2 system storages, you can configure synchronization to compare recordings.

A synchronization configuration is always created for 2 system storages. All recordings which are saved on one system storage are also copied to the other one and vice versa. That way, all recordings always exist on both system storages.

In a multi-core architecture, the system storage must not be synchronized between the Enterprise Cores.

Synchronization of the system storages is configured in the Servers module.

1. To create a synchronization configuration, click on the menu item *Servers* > *Manage Synchronization Configuration* in the toolbar of the main view.

Fig. 14: Menu item Manage Synchronization Configurations

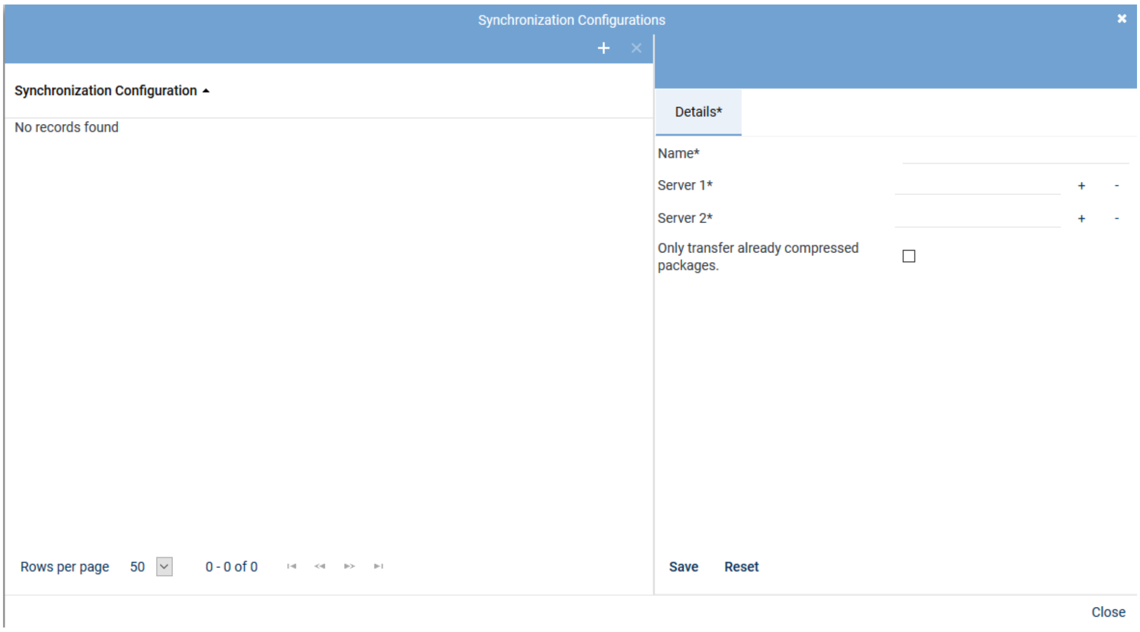

Fig. 15: Configure synchronization configurations

The following options are available:

+	*Create*	Creates a new synchronization configuration, see chapter "Create synchronization configuration", p. 20.
×	*Delete*	Deletes the selected synchronization configuration, see chapter "Delete synchronization configuration", p. 20.

A synchronization configuration becomes active upon saving it and remains active until deleted. During this time, both system storages are regularly checked for new content and synchronized.

A server which is already used in a synchronization configuration cannot be used in another synchronization configuration.

6.2.1 Create synchronization configuration

1. In the window *Administrate Synchronization Configuration*, click on the icon + *(Create)*.
   ⇨ The tab *Details* becomes active.

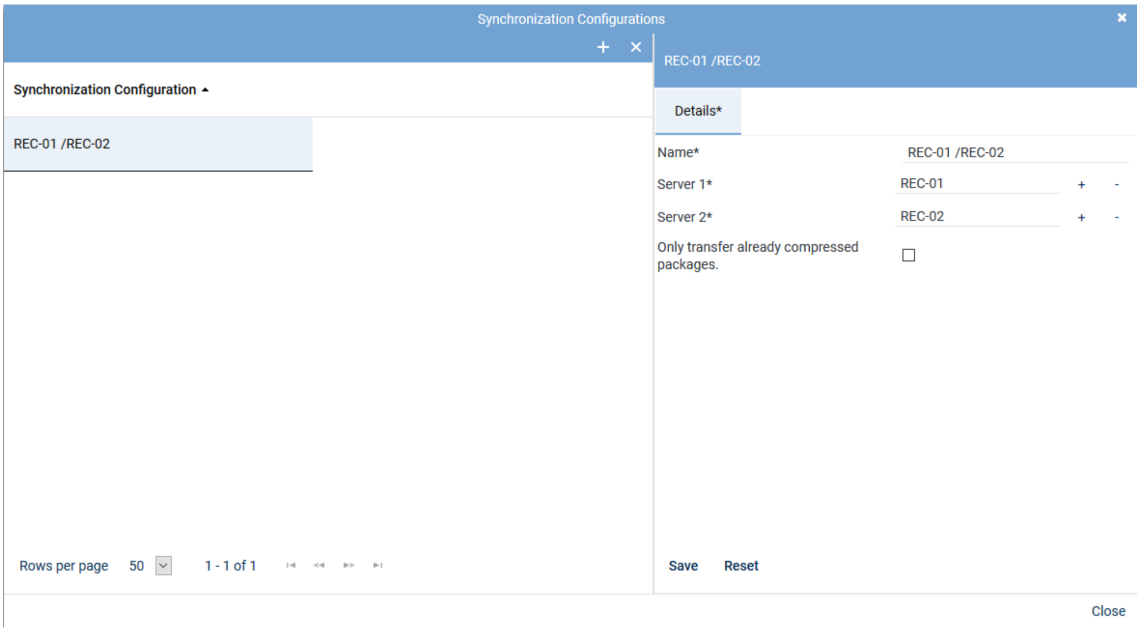

Fig. 16: Create synchronization configuration

2. Complete all fields for the new synchronization configuration:

Name	Enter a name for the synchronization configuration.
Server 1 / Server 2	Click on the button + next to the entry field to select the respective server for the synchronization of the system storage from the list of available servers. If you would like to delete an entry in one of the entry fields, click on the button - next to the respective entry field.
Only transfer already compressed packages	Select whether data which has not yet been compressed is supposed to be transferred, too. ☑ = Uncompressed data is transferred, too. ☐ = Only compressed data is transferred. **NOTICE!** This option is not available until you have entered and saved the two servers.

3. Click on the button *Save* to apply the configuration.
4. Click on the button *Close* to finish this configuration step and close the window.

6.2.2 Delete synchronization configuration

1. In the window *Administrate synchronization configurations*, select the synchronization configuration you would like to delete.
2. Click on the icon × *(Delete)* in the toolbar of the window.

⇨ The synchronization of the two entered system storages is finished.

⇨ The selected synchronization configuration is deleted.

7 Redundancy options

To ensure the unrestricted functionality of the recording system in the event of an error, you can install the following components redundantly:

- Application server

  See Multi-core system.
- Database

  See chapter "Redundant database instances", p. 29.
- Individual recording components

  See chapter "Redundant recording components", p. 31.
- Entire recording trunks (Recording Module Service, CTI*connect* Service, and Recording Control Service)

  See chapters:

  chapter "Architecture types for failover concepts", p. 11

  chapter "Architecture types for parallel recording", p. 14

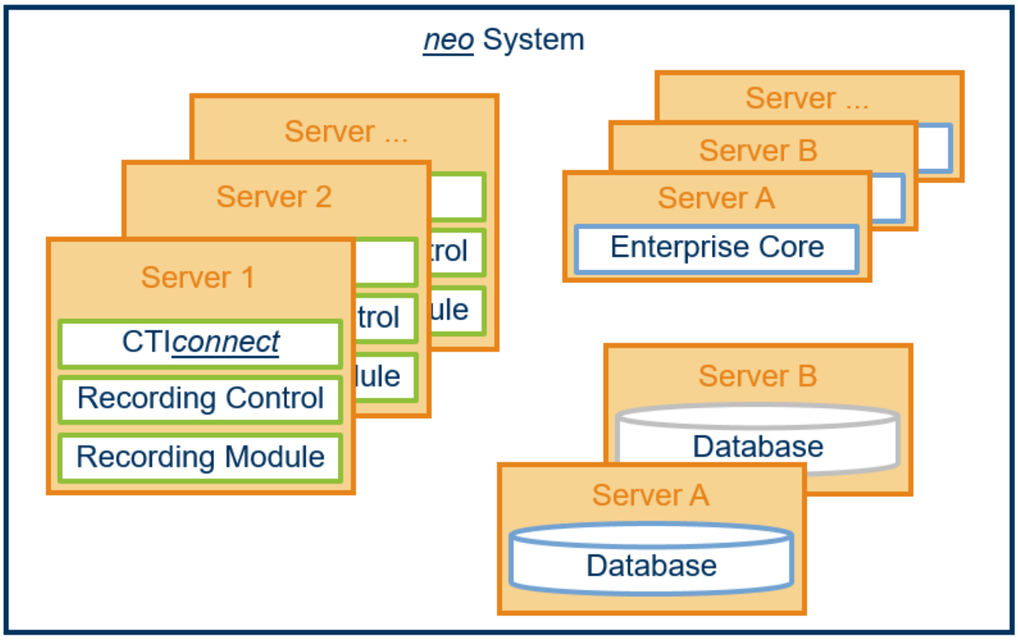

Fig. 17: Redundancy options

8 Possible system architectures

The ASC recording system supports different system architectures which results from the possibility to install the individual system components on different servers.

The installation routine of the ASC software automatically installs all software components required to run a server as recording server. Optionally, you can select during the installation of the ASC software whether a server should assume the function of an application server (app server) in addition to the function of a recording server or contain the database:

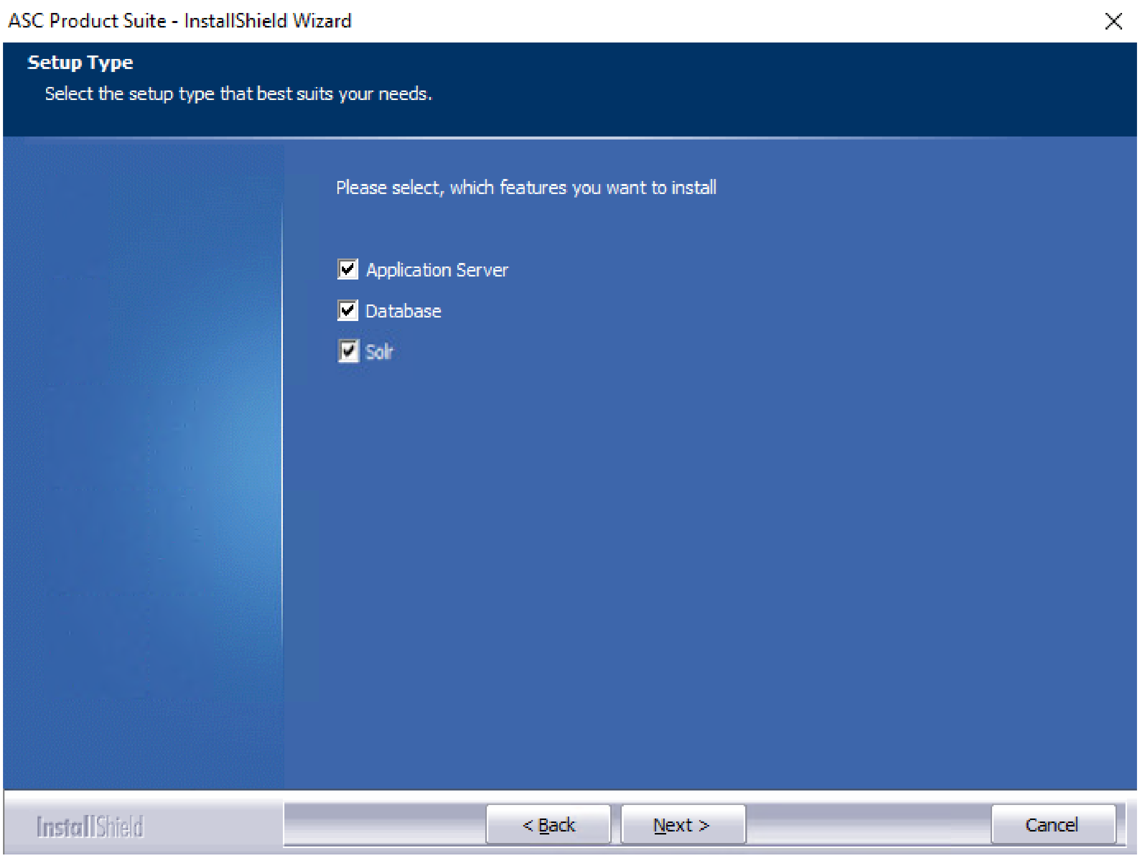

Fig. 18: Selection of optional software components

- *Application server*: All services relevant for the web applications are installed. The server can be used as app server.
- *Database*: The PostgreSQL database contained in the installation package is installed (internal database).
- *Solr*: The application contained in the installation package which enables full-text search in combination with INSPIRATION*neo* is installed.

Once the installation has been finished you can neither add missing components nor remove installed components.

For detailed information about the installation of the ASC software refer to the installation manual *Installation of the recording software of ASC*.

Basic system architectures

- Single-server system

  All software components including the database have been installed on one single server. See chapter "Single-server system", p. 23.
- Multi-server system

The individual software components have been installed on different servers. Different constellations are possible. However, each software component must have been installed at least once somewhere in the system.

Instead of the internal database, you can also use an external database.

See chapter "Multi-server systems", p. 23.

Supported redundancy options

- Multi-core system

  (Multi-server system with redundant app server)

  The *neo* recording software has been installed on several servers. The software components for the app server must have been installed on at least 2 servers.

  Instead of the internal database, you can also use an external database.

  See Multi-core system.

- Multi-server system with redundant recording components

  The individual software components have been installed on different servers. The recording components of a recording trunk may have been installed on one or on different servers. The individual recording components can be installed on several servers and integrated redundantly via the recording architecture.

  See chapter "Redundant recording components", p. 31.

- Failover function with an internal PostgrSQL database

  (Multi-server system)

  The *neo* recording software has been installed on several servers. When installing the *neo* software on different servers, the provided database software will be installed on 2 servers. One database is activated and another is configured as a standby.

  See chapter "Redundant database instances", p. 29.

8.1 Single-server system

With an architecture of this type, all recording components as well as the Enterprise Core and the database are located on one server and can thus only be installed once. There are no redundant recording components. A single-server system is always a single-core system with an All-in-one Basic recording architecture.

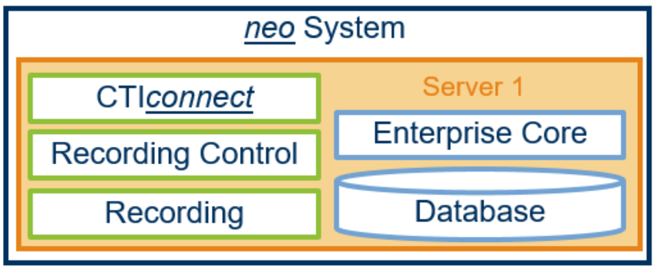

Fig. 19: Single-server system

8.2 Multi-server systems

In multi-server systems, the *neo* software is distributed among several servers can may thus be installed redundantly, too. There are a wide range of options regarding distribution and redundancies.

For the operation of a multi-server architecture, a network bandwidth of a minimum of 10 Mbit between the *neo* servers is required.

The network latency between Enterprise Core and database must be ≤ 10 milliseconds.

For the operation of a multi-core architecture, a Layer 4 Load Balancer is required. The Load Balancer must be provided by the system provider.

8.2.1 Exemplary installation

Multi-server system with 2 servers with All-in-one Recording, separate Enterprise Core and internal database

You can install the components of the recording architecture on one server and the app server components together with the database on a second server.

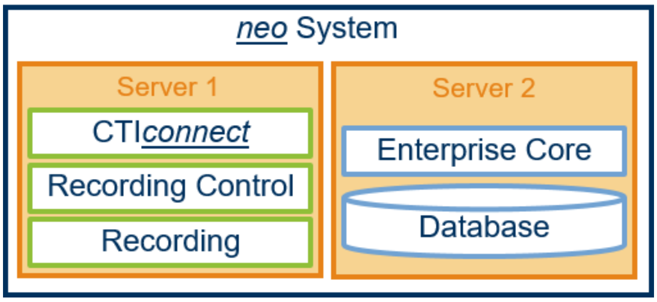

Fig. 20: System architecture with 2 servers and All-in-one Recording recording architecture

Multi-server system with 3 servers with All-in-one Recording, separate Enterprise Core and external database

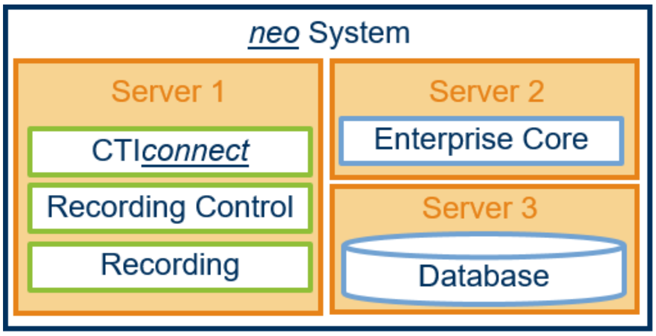

Fig. 21: System architecture with All-in-one Recording recording architecture, separate Enterprise Core and separate database

Multi-server system with 4 servers with parallel All-in-one Recording, corresponding Enterprise Core and external database with failover option

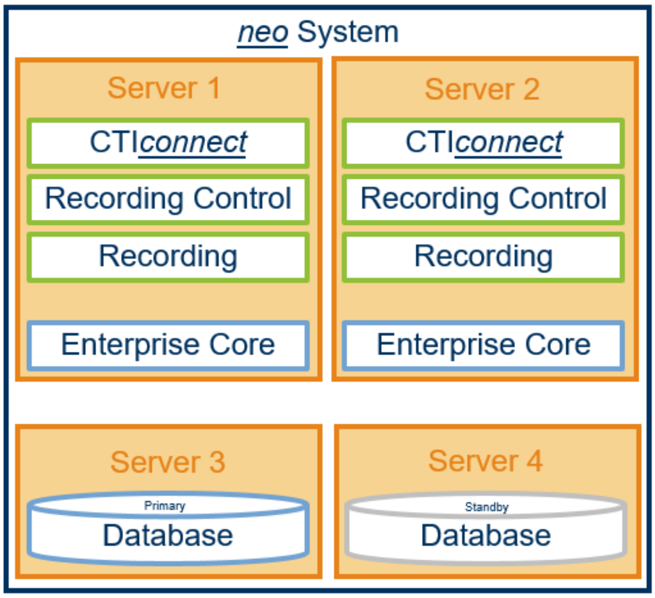

Fig. 22: System architecture with All-in-one Parallel Recording recording architecture and separate failover database

For this system architecture you must create a mutual synchronization job on server 1 and server 2. All recordings which are saved on one system storage are also copied to the other one and vice versa. That way, all recordings always exist on both system storages. If one of the two system storages fails, you can thus access the recordings of the failed system storage via the operative one.

For information about the configuration of synchronization jobs refer to the administration manual for system providers *Configuration servers and recording architectures*.

In addition, you can install individual recording components on separate servers.

For information about the configuration of servers and recording architectures refer to the administration manual for system providers *Configuration servers and recording architectures*.

Multi-server system with 4 servers, with All-in-one Failover Recording, redundant Enterprise Core and external redundant database

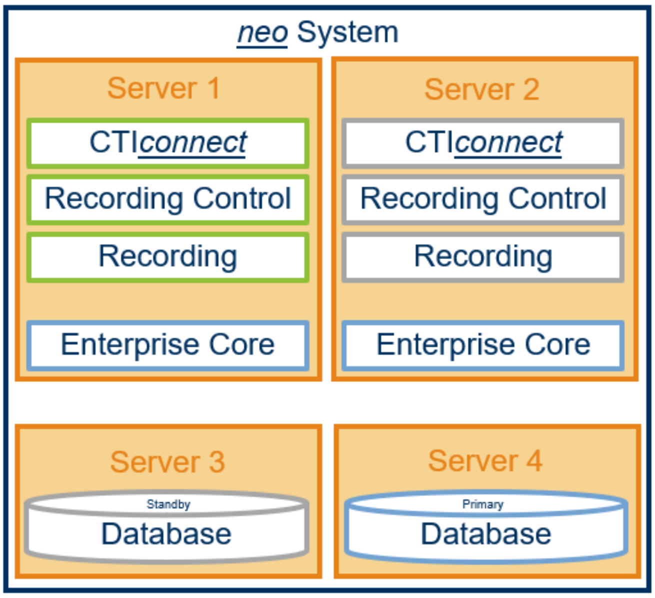

Fig. 23: System architecture with 4 servers, All-in-one Failover Recording recording architecture, redundant Enterprise Core and external redundant database

In failover recording, Enterprise Core and database can be set up redundantly, too. The two Enterprise Cores may be installed on the server with the recording components while the databases should run on their own separate servers.

Since enterprise core and database have been set up redundantly, the recorded conversations can be stored in the system at any moment and retrieved immediately. In case of the failure of a server, even configuration and other user interactions continue to be available. However, the complexity of the system's installation and maintenance increases.

Multi-server system with 7 servers, with parallel Multi-Server Recording, separate Enterprise Core and internal database

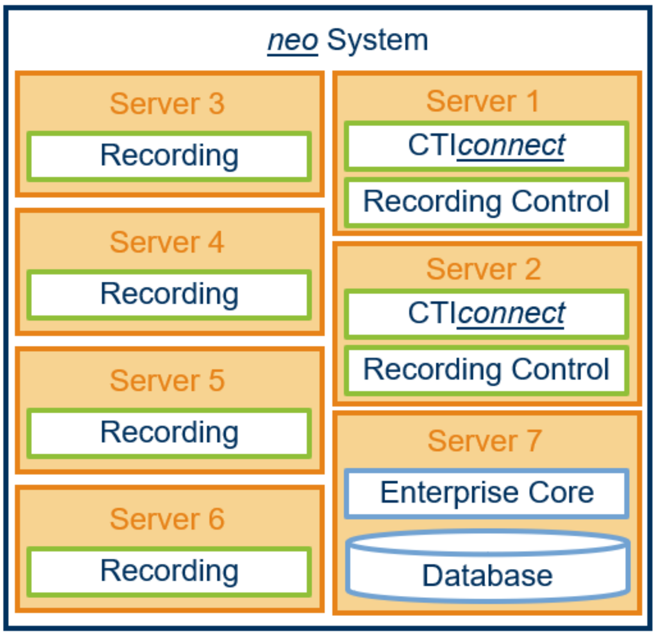

Fig. 24: System architecture with 7 servers, Multi-Server Parallel Recording recording architecture and separate Enterprise Core with database

The recording components on different servers can record different recording lines or can be configured for parallel recording as redundancy.

8.3 Single-core system

In this solution, there is only one Enterprise Core; it may be installed together with all other components on one server. Alternatively, the recording modules and the database may each be installed separately on other servers and set up redundantly, too. Single-core refers to one Enterprise Core only.

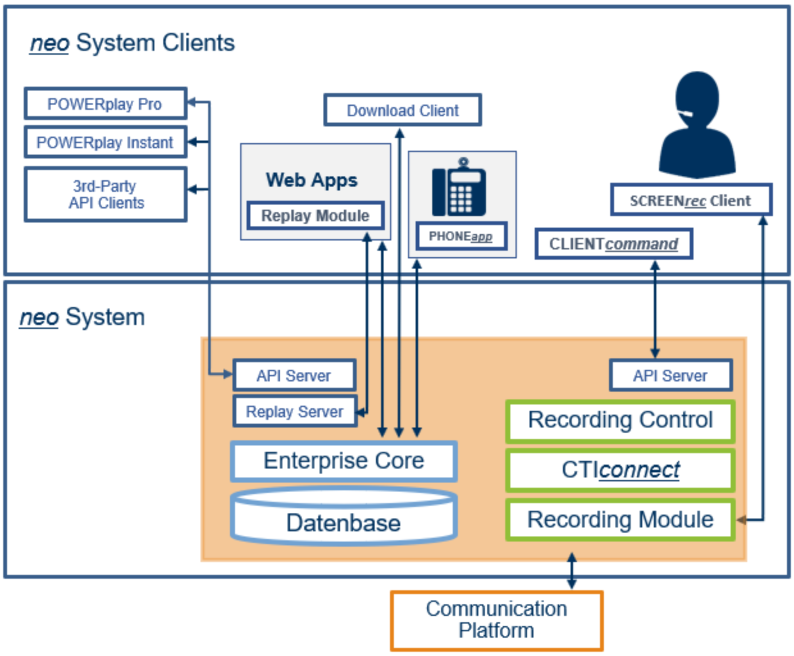

Fig. 25: Principle Single-server system with single core

8.4 Multi-server system with single core

In a multi-server system, you can install the Enterprise Core on a separate application server (app server):

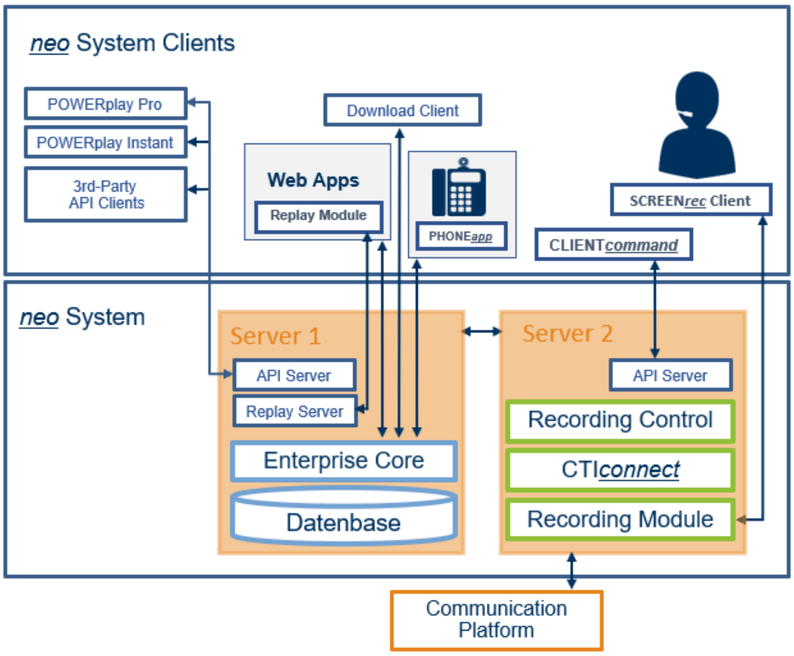

Fig. 26: Multi-server system with one Enterprise Core, internal database and All-in-one Recording

Furthermore, multi-server systems allow you to set up the following components redundantly for failover purposes:

- Recording components
- Databases

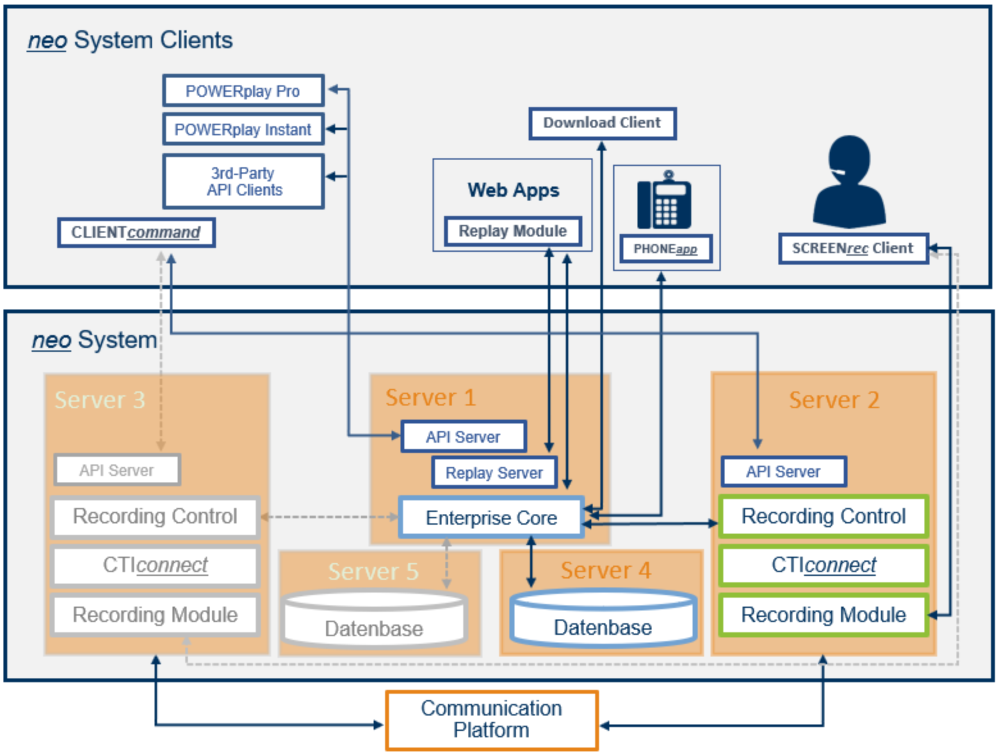

Fig. 27: Multi-server system with failover recording components and failover database

8.5 Multi-server system with multi-cores

A multi-core system consists of 2 or more servers on which the Enterprise Core has been installed (application server).

For load balancing or to safeguard recording in case of a failure of an application server (app server, you can set up several application servers in a server farm. In this server farm, the system load is distributed automatically among the different application servers. If one application server fails, the remaining application servers take over all tasks. All application servers use the same database.

During the installation of the recording software, you can configure which application servers are supposed to be available in your recording system to be deployed in the failover concept

To operate a multi-core architecture, a Layer 4 Load Balancer is required. The load balancer has to be provided by the system provider.

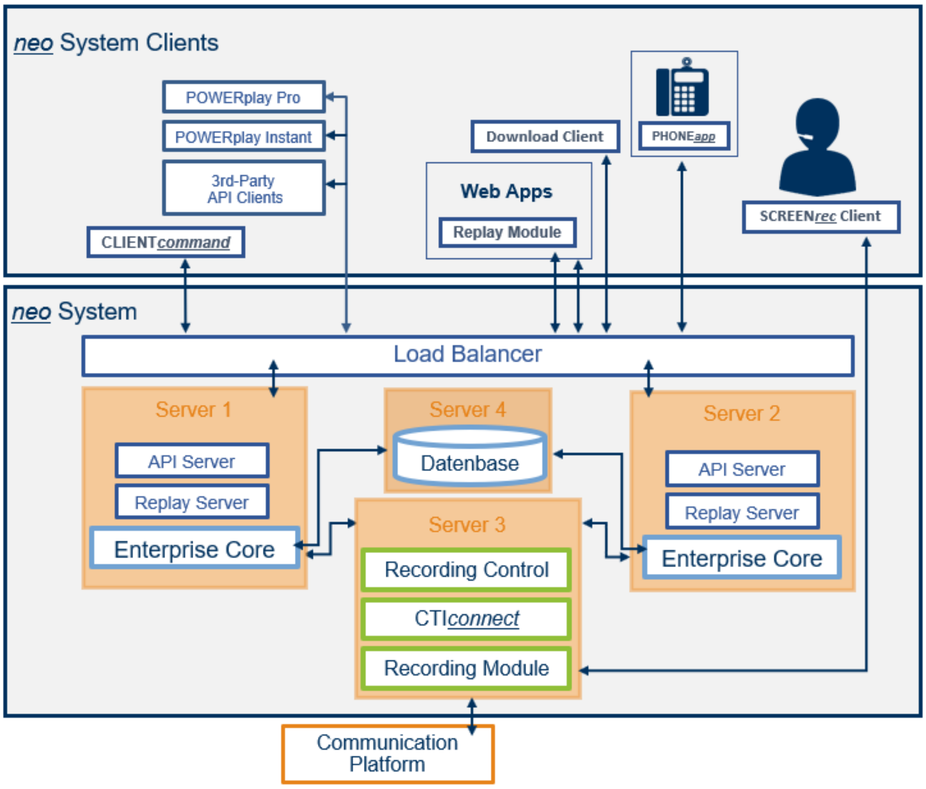

Fig. 28: Principle Multi-server system with multi-cores

ASC recommends additionally setting up the following components redundantly in multi-core systems to increase recording reliability and safeguard the access to the recordings:

- *Recording components*
- *Databases*

For information about the configuration of servers and recording architectures refer to the administration manual for system providers *Configuration servers and recording architectures*.

8.6 Redundant database instances

To secure the access to the recordings in case of a failure of the database, you can set up a second database instance.

What this failover solution looks like depends on the used database type:

PostgreSQL database

If you use an external PostgreSQL database, you can install a second PostgreSQL database instance on basis of the *neo* software and configure the system to copy the data from the primary database continually to the standby database. That way, both database instances contain the latest data at every moment. In case of an error, you can switch to the standby database manually.

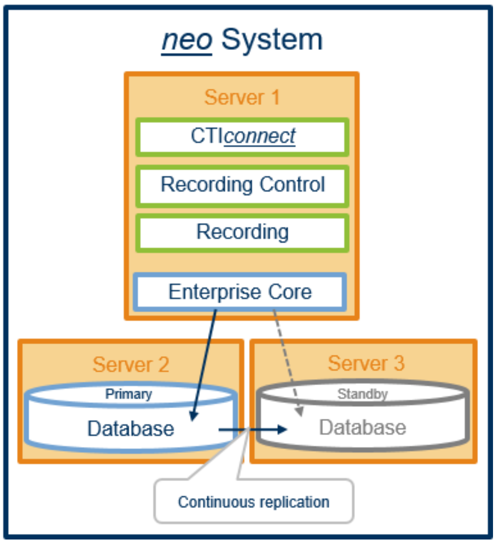

Fig. 29: Redundancy solution for PostgreSQL databases

ASC recommends installing the primary database as well as the standby database on a separate server each.

MSSQL database

If you use an MSSQL database, configure the redundant database according to the manual of the manufacturer.

For redundant MSSQL databases, we take the high availability of AlwaysOn Failover Cluster Instances for granted.

To secure access in case of a failure of the Enterprise Core, you can install the Enterprise Core redundantly on a second server, too.

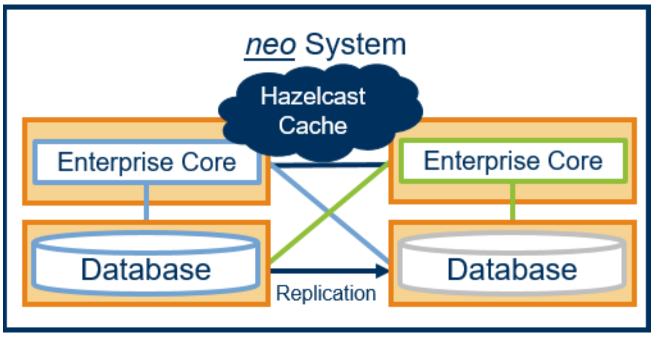

Fig. 30: Failover operation for 2 Enterprise Cores and redundant database

For redundant Enterprise Cores and redundant external databases, the license values are buffered in the Hazelcast cache. Via the virtual cache, the system status of the separate Enterprise Cores are aligned so that the operative Enterprise Core addresses the currently active database in case of an error.

For information about the update process of the respective system architecture refer to the installation manual for system provider *Software updates*.

The network latency between Enterprise Core and database must be ≤ 10 milliseconds, depending on the type of the database.

8.7 Redundant recording components

To safeguard recording in case of a failure of a recording server or a recording component, you can set up different recording architectures by means of the application System Configuration, in which selected or all recording components have been set up redundantly.

Examples:

Multi-Server Recording architecture with redundant Recording modules

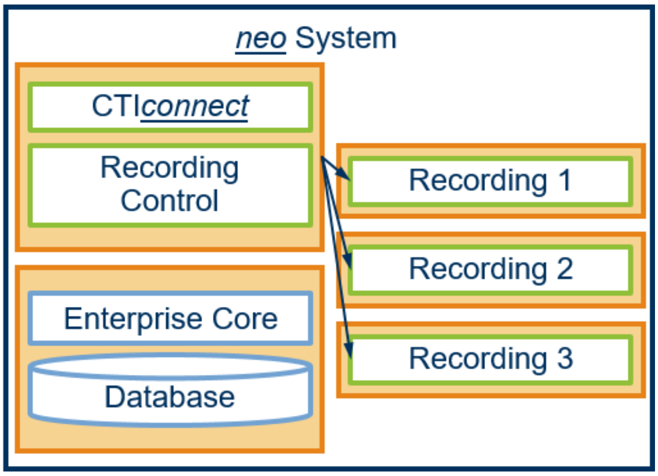

Fig. 31: Recording architecture with redundant Recording modules

Multi-Server Parallel Recording architecture with redundant Recording and Recording Control modules

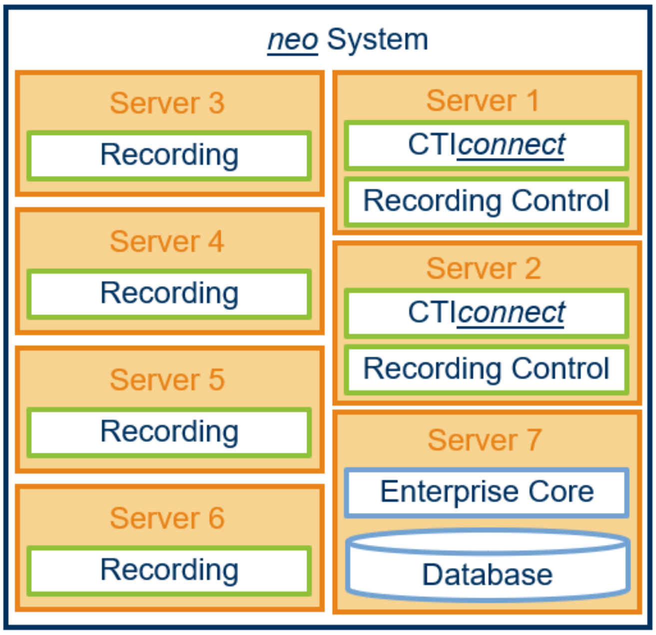

Fig. 32: Multi-Server Parallel Recording recording architecture with redundant Recording modules and Recording Control modules

For information about the configuration of failover recording architectures refer to the installation manual *Configuration servers and recording architectures*.

9 Recommended system architectures

9.1 Default architecture 1

All-in-one Basic recording architecture with 1 server

The default architecture 1 consists of 1 server with an *All-in-one Basic* recording architecture including Enterprise Core and an internal database.

Recommendation:

This is the ideal solution for small and medium installations which do not require a high-availability recording as this solution does not offer a redundancy.

In this solution, all components are installed on the same server. This includes an Enterprise Core and a database as well as an *All-in-one Basic* recording architecture.

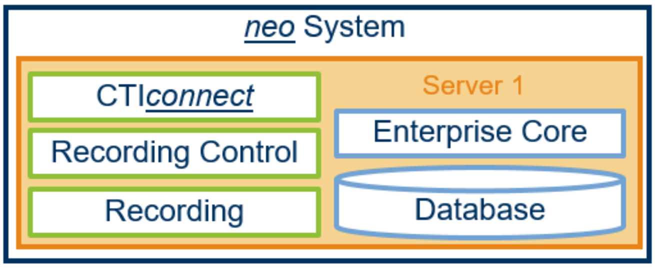

Fig. 33: Single-server system

9.2 Default architecture 2

All-in-one Basic recording architecture with 2 servers

Default architecture 2 consists of a system with 2 servers:

- *1 server with All-in-one Basic Recording*
- *1 Server with Enterprise Core and database*

Recommendation:

This is the ideal solution for medium installations which do not require a high-availability recording as this solution does not offer a redundancy.

In this solution, all recording components are installed on one server. Enterprise Core and database are installed on a second server to increase performance.

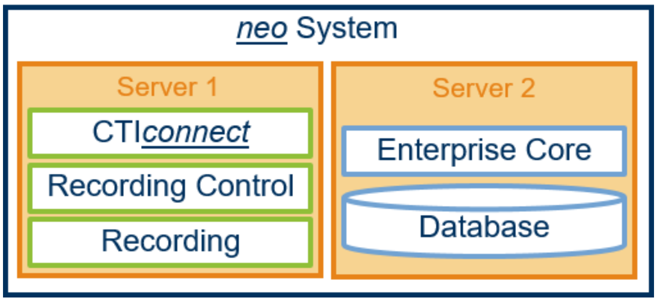

Fig. 34: System with 2 servers and 1 All-in-one recording architecture

The recordings may be replayed on server 1 or server 2:

1. When replaying recordings on server 1, a *replay server* must have been configured here.
2. When replaying recordings on server 2, a *data transfer* from server 1 to server 2 must have been configured. Data may only be transferred for replay or data storage purposes.

The network latency between all central ASC system components must not be higher than 10 milliseconds.

For information about the configuration of servers and recording architectures refer to the administration manual for system providers *Configuration servers and recording architectures*.

9.3 Default architecture 3

All-in-one Failover recording architecture with 2 servers

Default architecture 3 consists of a system with 2 servers:

- *1 server with All-in-one Failover*
  including active Enterprise Core and internal active database
- *1 server with All-in-one Failover*

Recommendation:

This is the ideal solution for small and medium installations which require a redundant recording which bear with minor losses while switching to architectures.

In this solution, the recording components and the Enterprise Core are installed along with the database on the first server. On a second server, only the recording components are installed once again. The recording components on server 2 serve as the primary recording architecture. The recording components on server 1 serve as failover. If a component relevant for recording fails, the system activates the standby component and recording continues on server 1.

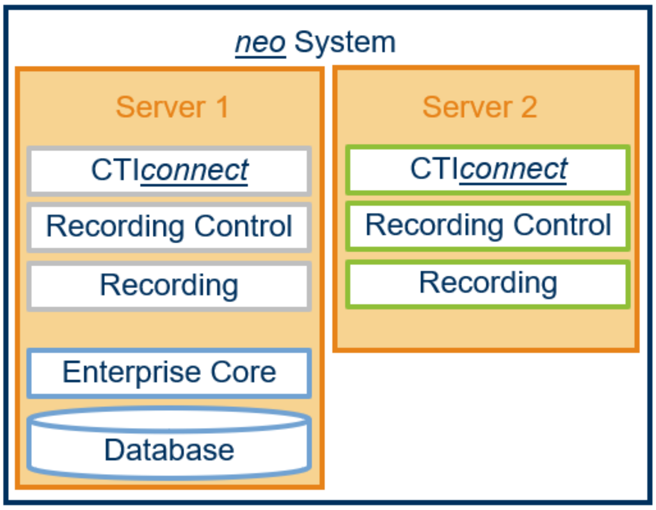

Fig. 35: System with 2 servers and All-in-one Failover

ASC recommends to set up a data transfer from server 2 to server 1, as server 1 will then contain all recordings even if server 2 fails. If data is stored on a storage expansion, a data transfer is not required.

The network latency between all central ASC system components must not be higher than 10 milliseconds.

After a failover case, you must switch back to server 2 manually.

For information about the configuration of servers and recording architectures refer to the administration manual for system providers *Configuration servers and recording architectures*.

9.4 Default architecture 3 plus screen

All-in-one Failover recording architecture with 4 servers

Default architecture 3 plus screen consists of a system with 4 servers:

- *1 server with All-in-one Failover including active Enterprise Core and internal active database plus failover recording components as part of an All-in-one Failover recording architecture, (3) failover*
- *1 server with All-in-one Failover recording components, (1) active recording*
- *2 servers with All-in-one Failover recording components for screen recording, (2) active, (4) failover*

Recommendation:

This is the ideal solution for small and medium installations which require a redundant recording which bear with minor losses while switching to architectures.

In this solution, the audio recording components and the Enterprise Core are installed along with the database on the first server. On the second, third, and fourth server, only the recording components are installed once again. The recording components on server 2 and 3 serve as the primary recording architecture for audio and screen. The recording components (3) and (4) on servers 1 and 4 serve as failover. When a component relevant for recording fails, the system activates the standby components and recording continues on the recording components (3) and (4) on servers 1 and 4.

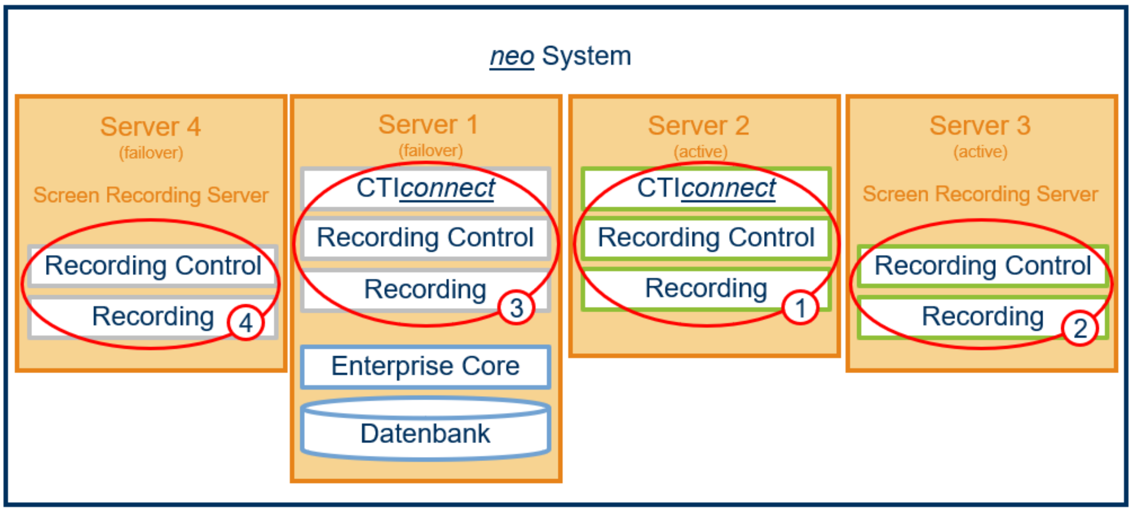

Fig. 36: System with 4 servers, All-in-one Failover plus screen recording

ASC recommends to set up a data transfer from servers 2, 3, and 4 to server 1, as server 1 will then contain all recordings even if server 2 fails. If data is stored on a storage expansion, a data transfer is not required.

The network latency between all central ASC system components must not be higher than 10 milliseconds.

After a failover case, you must switch back to server 2 manually.

For information about the configuration of servers and recording architectures refer to the administration manual for system providers *Configuration servers and recording architectures*.

9.5 Default architecture 3 a

All-in-one Failover recording architecture with 4 servers

Default architecture 3 a consists of a system with 4 servers:

- *2 servers with All-in-one Failover and an active Enterprise Core*
- *1 server with the external active database*
- *1 server with a standby database*

Recommendation:

This is the ideal solution for installations which require the recording to be switched to a failover system in case of a failure. The Enterprise Core of both systems runs redundantly on a permanent basis. If a database fails, the system can fall back on a redundant database.

In failover recording, Enterprise Core and database can be set up redundantly, too. The two Enterprise Cores may be installed on the server with the recording components while the databases should run on their own separate servers.

Since enterprise core and database have been set up redundantly, the recorded conversations can be stored in the system at any moment and retrieved immediately. In case of the failure of a server, even configuration and other user interactions continue to be available. However, the complexity of the system’s installation and maintenance increases.

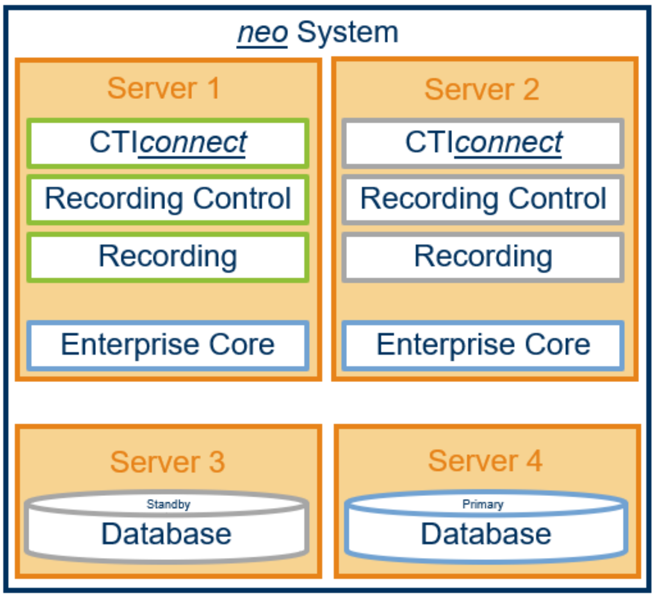

Fig. 37: System with 4 servers, All-in-one Failover Recording, redundant Enterprise Core and external redundant database

ASC recommends to set up a data transfer from server 1 to server 2, as server 2 will then contain all recordings even if server 1 fails. If data is stored on a storage expansion, a data transfer is not required.

The network latency between all central ASC system components must not be higher than 10 milliseconds.

For information about the configuration of servers and recording architectures refer to the administration manual for system providers *Configuration servers and recording architectures*.

9.6 Default architecture 3 a plus screen

All-in-one Failover recording architecture with 6 servers

Default architecture 3 a consists of a system with 6 servers:

- *2 servers with All-in-one Failover recording components (1) and (2) and an active Enterprise Core each*
- *2 servers with All-in-one Failover recording components for screen recording only with the recording module (3) and (4)*
- *1 server with the external active database*
- *1 server with an external standby database*

Recommendation:

This is the ideal solution for installations which require the recording to be switched to a failover system in case of a failure. The Enterprise Core of both systems runs redundantly on a permanent basis. If a database fails, the system can fall back on a redundant database.

In failover recording, Enterprise Core and database can be set up redundantly, too. The two Enterprise Cores may be installed on servers 1 and 2 with the audio recording components while the databases and screen recording with recording modules (3) and (4) should run on their own separate servers.

In geo-redundancy scenarios, the standby database must be installed at the location where active recording has been installed. The active database must be installed at the location of the standby recording components.

Since enterprise core and database have been set up redundantly, the recorded conversations can be stored in the system at any moment and retrieved immediately. In case of the failure of a server, even configuration and other user interactions continue to be available. However, the complexity of the system's installation and maintenance increases.

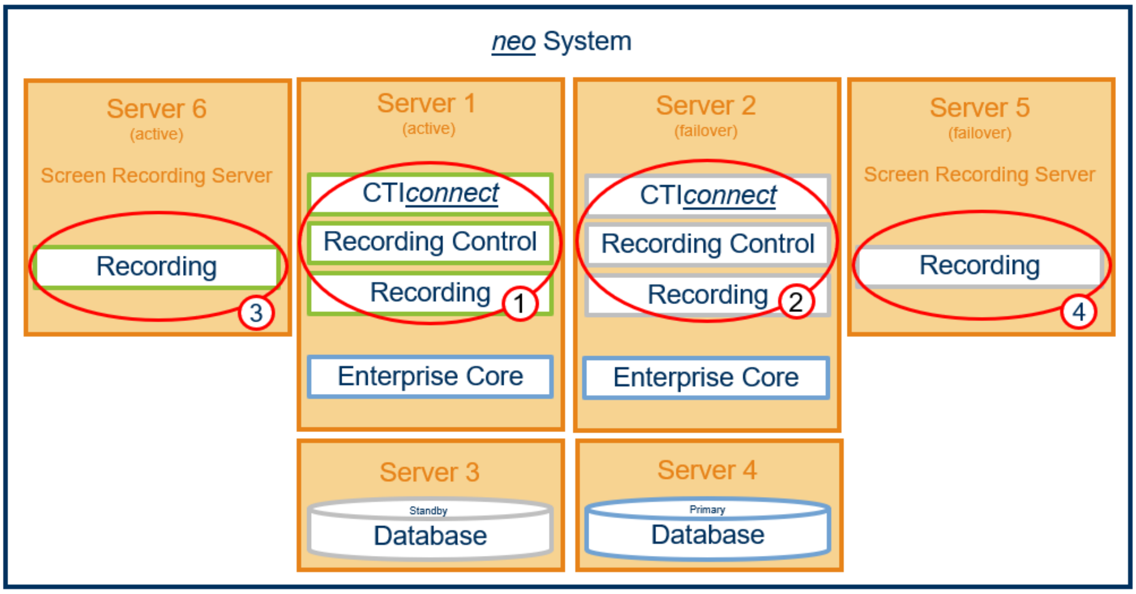

Fig. 38: System with 6 servers, All-in-one Failover recording, All-in-one Failover for screen recording, redundant Enterprise Core, and external redundant database

The network latency between all central ASC system components must not be higher than 10 milliseconds.

For information about the configuration of servers and recording architectures refer to the administration manual for system providers *Configuration servers and recording architectures*.

9.7 Default architecture 3 b

Active-Active Recording architecture

This system architecture is set up in *neo* with All-in-one Parallel Recording architectures on 2 servers.

Default architecture 3 b consists of a system with 2 servers:

- *1 server with All-in-one Parallel Recording*
  including active Enterprise Core and internal active database
- *1 server with All-in-one Parallel Recording*

This recording architecture is only available for the recording variants SIPREC, SIP active, and Cisco UCM active.

Description of Active-Active Recording

In this parallel recording solution, a set of 2 identical servers with the recording components are active at the same time. This means that the recording solution can accept recordings on any server. The communication platform of the customer does not have to provide 2 parallel streams for recording but can send the recordings in round-robin principle to an available recording server (load balancing). On top of that, this architecture can be used as hot-standby solution. If a recording server fails, new recordings can be sent to the second recording server immediately. The main advantage of this setup is the high availability without the need to record everything twice. A disadvantage when compared with a dual-stream parallel solution is that the active recordings on the recording server are stopped in case it fails.

Recommendation:

This is the ideal solution for small and medium installations which require a redundant recording and which strive to avoid loss of recordings while switching to architectures.

In this solution, the recording components and the Enterprise Core are installed along with the database on the first server. On a second server, only the recording components are installed. The recording components on both servers run in a parallel recording architecture and are both active.

The PBX or the SBC distributes the data to the recording servers in 2 different variants, though:

- Round robin

  In round-robin principle, the opposite side sends data alternately to one of the two servers. If one server should not be available, the data is sent to the other server.

- Load balancing

  In load balancing principle, the servers receive a priority. The opposite side first sends the data to the prioritized server; if this server should not be available, the data is sent to the server with the next lower priority.

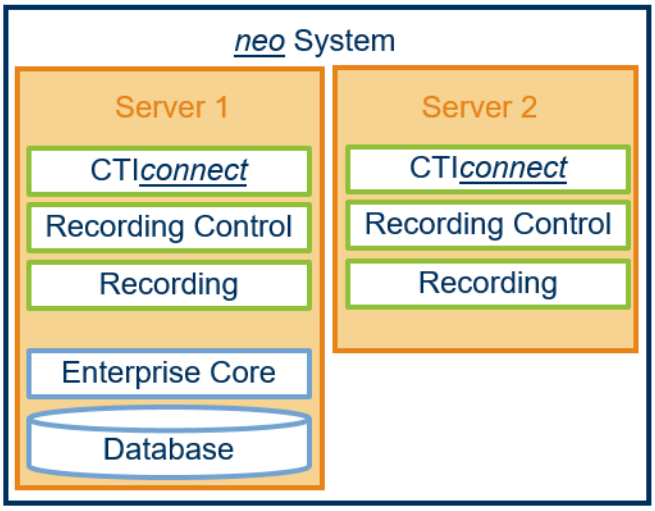

Fig. 39: System with 2 servers and All-in-one Parallel Recording

The network latency between all central ASC system components must not be higher than 10 milliseconds.

For information about the configuration of servers and recording architectures refer to the administration manual for system providers *Configuration servers and recording architectures*.

9.8 Default architecture 4

All-in-one Parallel Recording recording architecture with 3 servers

Default architecture 4 consists of a system with 3 servers:

- *2 server with All-in-one Parallel Recording*
- *1 Server with Enterprise Core and database*

Recommendation:

This is the ideal solution for installations which require full recording availability at any moment and which have to avoid losses of recordings at any cost.

In this solution, all components relevant for recording are installed as All-in-one Parallel Recording on two separate servers each. Recording takes place in parallel on server 1 and server 3. In case of a failure, there is no need to switch to another architecture and recording can be guaranteed without interruption. Besides that, the Enterprise Core is installed along with the database on a third server. Neither of them is redundant but can be expanded accordingly.

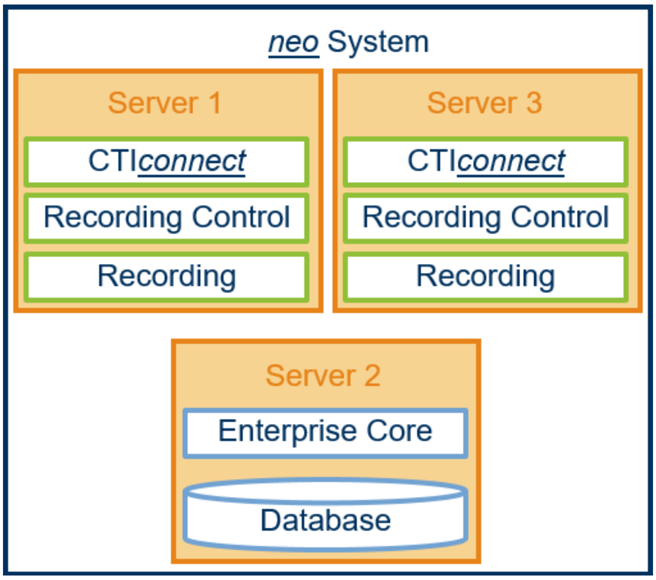

Fig. 40: System with 3 servers and All-in-one Parallel Recording

To have all recordings available, ASC recommends to synchronize the 2 recording servers that are running in parallel. Alternatively, you can transfer all recordings from server 1 and 3 to server 2 and replay them there.

If data is stored on a NAS, a data transfer to the servers is not necessary.

The network latency between all central ASC system components must not be higher than 10 milliseconds.

For information about the configuration of servers and recording architectures refer to the administration manual for system providers *Configuration servers and recording architectures*.

9.9 Default architecture 4 a

All-in-one Parallel Recording recording architecture with 4 servers

Default architecture 4 a (previously 4+) consists of a system with 4 servers:

- *2 servers with All-in-one Parallel Recording including redundant Enterprise Core*
- *1 server with the external active database*
- *1 server with a standby database*

Recommendation:

This is the ideal solution for installations which require full recording availability at any moment and which have to avoid losses of recordings at any cost.

In this solution, all components relevant for recording are installed separately as All-in-one Parallel Recording including the Enterprise Core on two separate servers each. Recording takes place in parallel on server 1 and server 2. In case of a failure, there is no need to switch to another architecture and recording can be guaranteed without interruption. In addition, the active database is installed on a third server. On a fourth server, the standby database is installed.

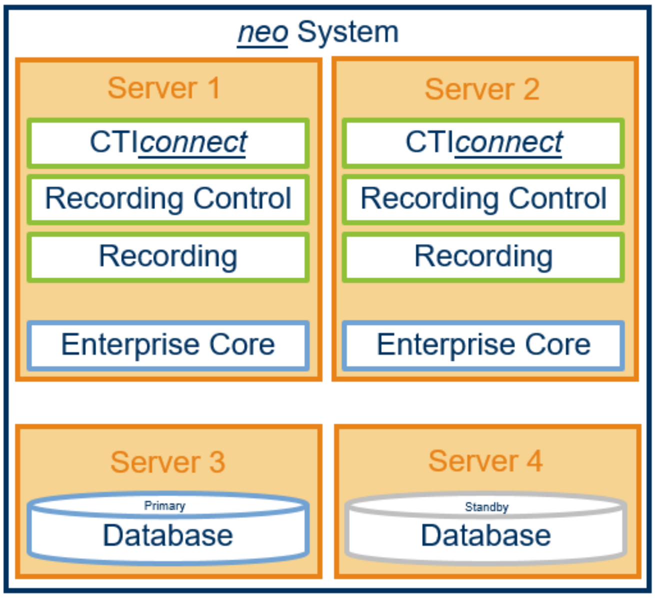

Fig. 41: System with 4 servers and All-in-one Parallel Recording

The network latency between all central ASC system components must not be higher than 10 milliseconds.

For information about the configuration of servers and recording architectures refer to the administration manual for system providers *Configuration servers and recording architectures*.

10 Definition of the terms

10.1 System, general

Recording architecture

Composition of all required recording components.

- Recording Control

  This service controls the recording according to the recording plan.
- CTI*connect* (optional)

  This service receives additional data about the recordings from the PBX.
- Recording Module

  This service creates the recording data.

A recording architecture defines in which way these recording components interact and on which servers the individual recording components are activated.

Single-core system

Recording system in which the Enterprise Core has been installed on one single server.

In a multi-server system, this may be a separate server (application server). In a single-server system, the Enterprise Core has been installed on the same server as the other recording-relevant components.

Multi-core system

Recording system in which the Enterprise Core has been installed and is used on several servers. The Enterprise Core may be installed on separate servers (application server) or along with the other recording-relevant components.

Single-server system

Recording system in which all components (such as Enterprise Core, recording components, database) have been installed on the same server.

Multi-server system

Recording system in which the individual components (such as Enterprise Core, recording components, database) have been installed on different servers.

- The functionalities of the application server and of the recording server are installed on one server. The database is installed on a second server.
- The functionalities of the application server, of the recording server as well as the database have been installed or activated on their own server.
- The system uses several application servers (multi-core system), a recording server and a server for the database.

Tenant

This term is to be understood in the sense of a technical structure and not a synonym for “client”. A tenant can, for instance, be a department, a company or a group of employees with their own data that only the respective department, company or group has access to.

Each tenant can create its own employees as users, its own administrators and agents in the recording system.

Every *neo* system is initially installed as a 1-tenant system with one predefined tenant, the 1st-tenant. The system provider is set up as tenant, too. However, the system provider is not another tenant in the true sense of the word.

In multi-tenant systems, the system provider can create additional tenants.

1-tenant system

In a 1-tenant system, there is only the tenant which has automatically been created during the installation besides the system provider. The system provider cannot create other tenants.

Multi-tenant system

In a multi-tenant system, the system provider can create additional tenants besides the tenant which has automatically been created during the installation.

Multi-channel recording

Multi-channel recording means that different communication channels such as audio, video, chat can be recorded.

System provider

Operator of the recording system. The system provider is responsible for the basic administration and maintenance of the recording system and for the configuration of the functions that all tenants are supposed to be able to use. In multi-tenant systems, the system provider additionally administrates the accounts of different tenants.

The system provider can create own employees as users as well as administrators in the recording system but no agents, though.

Reseller

A reseller has a restricted set of rights compared with those of a system provider or tenants.

- A reseller can create, delete, and administrate subordinated tenants and resellers.
- A reseller can create own employees as system users and administrate and delete them.

NOTICE! A reseller has no access to the user data of the individual tenants. Only the tenant itself can view and edit tenant-specific data.

10.2 Servers, types, and functionalities

Server

The term “server” is not necessarily restricted to hardware but may extend to services or functionalities which have been installed and activated on hardware.

Example:

The entire *neo* software including the app server components has been installed on a server. This server can be used as an application server. Since all other components relevant for recording have been installed on this server, too, it can additionally be used as recording server. By means of the user interface of the application System Configuration, further functionalities can be activated on the server. If e. g. the function “Replay” is activated, then the server serves as replay server, too. The server thus serves as application server, recording server, and replay server at the same time.

Application server

The application server (app server) is the server on which the Enterprise Core and the GlassFish software have been installed. Application servers can be set up redundantly in the system (multi-core system).

It is activated during the installation of the ASC recording software by activating the option *Application server*.

API server

The API server denominates the API service.

- The API server is an interface for the internal modules and for client applications.
- The API server is responsible for replay by means of the web browser. Not until the ASC API Server has started, can the replay server be activated and the corresponding API server assigned for replay in the web applications.

It is activated in the application *System Configuration* > *Servers module* > *tab Usage* > *group field API Sever* by activating the option API server and entering a name for the API server.

Recording server

The recording server is the server on which the conversations are recorded and saved. By using a multi-server, failover, or parallel recording architecture, recording servers can be set up redundantly in the system.

It is activated in the application *System Configuration* > *Recording Architectures module* > *tab Server Assignment* by assigning the server to a recording architecture and selecting the recording type.

Authentication server

The authentication server is either the server on which the Dongle Manager is running or the server on which the LMS (ASC License Management Service) can be reached.

Data storage server

The data storage server serves to store recordings.

It is activated in the application *System Configuration* > *Servers module* > *tab Usage* > *group field Data Processing* by activating the option *Data storage* > *Transfer data for data storage* and entering the data storage server as target server.

The server receives and saves the transferred recording data.
In the *Servers module* > *tab Usage*, you can see from which servers the data storage server receives data.

Database server

The database server is the server on which the database has been installed. In the database, the configuration of the recording system (settings in the different applications of the *neo* Suite) and the additional data of the recorded conversations are saved. Depending on the deployed database type, different redundancy solutions can be implemented.

Installing internal database

- Installed during the installation of the *neo* software by installing the included PostgreSQL database.

Installing external database

- An external database supported by ASC can be installed on a separate server.
- The connection to the database server is configured during the installation of the *neo* software.

Replay server

The replay server is a server on which the replay function has been activated which can thus replay recordings by means of the integrated replay feature. Only the data which has been recorded directly on this server or which has been transferred to this server for data storage or replay purposes is available for replay. The client computers of the system can connect to a replay server for replay purposes. Several replay servers can be created in a system.

They are activated in the application System Configuration > Servers module > *tab Usage* > *group field Replay* by activating the function *Replay* and entering a name.

By means of the different replay applications of the recording system, the client applications can connect with the server and access the recordings there for replay purposes.

For detailed information about the configuration of servers refer to the installation manual *Configuration servers and recording architectures*.

10.3 Recording types

Conversation

Umbrella term for the different types of communication that can be recorded. Conversation is used when there is no need to differentiate between different conversation and media types.

Conversation type

Type of communication, e. g. call, chat or SMS.

The recorded data may be conversations of different types:

Description	Icon	Conversation type	Recording format
Mere calls		Call	Audio
Mere screen recording		Work item	Screen video
Calls with screen recording		Call	Audio via phone, and screen video
Calls with video		Call	Audio and video
SMS		SMS/SDS	SMS/SDS text
SDS		SMS/SDS	SMS/SDS text
Chat messages		Chat	Chat text

Tab. 1: Conversation types

Call, chat, text message and *work item* are differentiated in recording.

- *Call*: Conversation by phone. Any combination of call and video recording can be selected.
- *Chat*:
  Conversation on a chat platform.
- *Text message*:
  Conversation on a short message service (SMS) or short data service (SDS).
- *Work item*:
  Screen activity **without** a reference to a call.

Session

Recorded conversations are processed as sessions in INSPIRATION*neo*. A session is the section of a conversation in which a certain agent is active. Precondition for a session is that the Recording Planner in the System Configuration has been activated. Sessions are conversations with screen recording, mere screen recordings (work item), conversations with video recording (video call), SMS/SDS (text messages), chats or mere call recordings. A session can consist of just a recorded phone call of an agent or additionally contain corresponding screen activity. Users can assign sessions to agents, filter sessions according to different criteria and reduce their number to a manageable amount for analysis or evaluation purposes.

Differences conversation and session

In general, you have to distinguish between *conversation* and *session*.

- *Conversation*
  Conversation refers to the entire call from the moment it is answered to the end of the call, regardless of internal transfers. If a consultation is initiated, though, then the consultation is a conversation of its own.
- *Session*
  A session is the section of a conversation in which a certain agent is active. Pausing the call (e. g. because of a consultation), does not finish this session. The session is not divided into 2 sessions. The consultation or transmitting creates at least one new session of its own. For each involved agent who is supposed to be recorded according to the recording plan a separate session is created. A session is always a recording section that refers to a certain agent. Consequently, the sections of a transferred conversation in which several agents have thus participated are displayed as a session for every agent but with different content.

10.3.1 Recording

10.3.1.1 EVOIPneo

EVOIP*neo* is a voice documentation system with a powerful scalable platform. The system can be deployed as a stand-alone recorder or in combination with several servers across different locations.

EVOIP*neo* is a sophisticated recording technology to comply with legal regulations such as MiFID II or Dodd Frank. EVOIP*neo* provides multi-channel recording of voice, screen, video, and chat.

10.3.1.2 EVOLUTIONneo

EVOLUTION*neo* is an addition to integrate traditional telephony into an ASC recording system. Besides VoIP recording, EVOLUTION*neo* offers interfaces to all standard TDM-based PBXs by means of dedicated recording cards in a server housing designed for this purpose.

Definition of names:

EVOLUTION*neo* is used for the entire EVOLUTION*neo* product family (EVOLUTION*neo*, EVOLUTION*neo* XXL and EVOLUTION*neo* eco). In case of differences or peculiarities of the individual systems, the entire product name will be indicated for the sake of clarity.

10.4 Drive categories

During their configuration, drives must be assigned to a category. This category defines what purpose the drive is to serve.

There are the following different categories:

1. **System storage**

   Drives that are used as system storage are created and configured during the installation. System storages are used exclusively for the recording of conversations. A system storage can neither be used for archiving nor for export or as storage expansion.

   There is exactly 1 system storage per server. All other drives can only be configured as storage expansion or data drive.

   You can save recordings in the system storage in compressed form.

2. **Storage expansion**

   Storage expansions serve to expand the system storage. The capacity of a storage expansion must be 10 % higher at least than the capacity of the system storage.

For every system storage, any number of storage expansions can be configured. To share a storage expansion to be used, you must first assign at least 1 tenant to the storage expansion. This implies that you can use as many active (shared) storage expansions per system storage as there are tenants in the system.

The recordings of all tenants assigned to a storage expansion are copied to this storage expansion. That way, the local availability of the recordings continues to be given for this tenant even if recordings are deleted from the system storage for capacity reasons. The recordings of tenants not assigned to a storage expansion are saved exclusively in the system storage.

You can assign any number of tenants to a storage expansion.

A storage expansion can neither be used for archiving nor for import or export.

3. **Data drive**

   A data drive is not used for the recording of conversations. Data drives can be used exclusively for archiving, import, and export.

   You can configure any number of data drives.

4. **Database drive**

   Exclusively the database has been installed on the database drive. You cannot install any other software components on this drive. The database drive is configured during the installation of the *neo* software if you are not using an external database. A maximum of 1 database drive can be configured per recording system.

5. **Application drive**

   The *neo* software has been installed in the application drive. The application drive is configured during the installation of the *neo* software. The drive where Windows has been installed is considered an application drive, too. Application drives can be used as source drive for the import of conversations.

6. **Recording Insights Transfer**

   The drive that is used for export and import of data from Azure Blob Storage where the recordings from Recording Insights are saved.

7. **Cisco Webex drive**

   This virtual drive is used to save and transfer recordings from Cisco Webex.

In virtual environments, you can exclusively use network drives for archiving, import, and export of data. Internal or USB drives are not supported as performance issues may occur when trying to access a drive that is no longer available.

List of figures

List of tables

Glossary

API

Application Programming Interface

API server

Server on which the API service runs. (API=Application Programming Interface)

App server

Application server or web server. In the system architectures: the server on which the Enterprise Core and the GlassFish software have been installed.

Communication platform

Umbrella term for the different sources that the communication to be recorded can come from (e. g. PBXs, chat servers, ...)

Load Balancing

In load balancing principle, the servers receive a priority. The PBX first sends the data to the prioritized server; if this server should not be available, the data is sent to the server with the next lower priority.

MiFID II

Markets in Financial Instruments Directive; directive for the harmonized regulation for investment services across the member states of the European Economic Area.

Multi-channel recording

signifies that the recording system can record different communication types, e. g. audio, video, SMS, or chat.

Multi-core system

Recording system in which several application servers (Enterprise Core) are used.

Multi-server system

Recording system in which the individual components (Enterprise Core, recording components, database) are installed on different servers.

NAS

Network Attached Storage is a file-level computer data storage server connected to a computer network providing data access to other devices on the network. NAS is usually used to provide independent storage capacity in a computer network without major effort. (Source: Wikipedia 4th May 2017)

PBX

Private Branch Exchange

Recording server

Server that the Recording Module service runs on. This service creates the recording data. A Recording system can contain one or several recording servers.

Replay server

Server on which the replay function has been activated. Recordings can be replayed via this server.

Round robin

In round-robin principle, the PBX sends data alternately to one of the available servers. If one server should not be available, the data is sent to another server.

SBC

Session Border Controller

SDS

Short Data Service (TETRA), text message

Single-server system

Recording system in which all components (Enterprise Core, recording components, database) are installed on the same server.

SIP

Session Initiation Protocol

SIPREC

Session Initiation Protocol Recording

SMS

Short Message Service, text message (GSM, landline)

TDM

Time Division Multiplexing is an umbrella term for time-slot-oriented interfaces, ITU G.703 defined. The term is used ASC-wide representative for conventional telephony.

USB

Universal Serial Bus

Video recording

A video recording can consist either of a screen video or of any other video.

VoIP

Voice over IP