

# **neo system architectures**



## **Installation manual for system providers**

6/22/2020

### **Product line neo, version 6.x**

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

EVOIPneo

EVOLUTIONneo / XXL / eco

EVOflex (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>.

Copyright © 2019 ASC Technologies AG. All rights reserved.

Windows is a registered trademark of Microsoft Corporation. VMware® is a registered trademark of VMware, Inc. All other marks and names mentioned herein may be trademarks of their respective companies.



## Contents

<b>1</b>	<b>General information .....</b>	<b>5</b>
<b>2</b>	<b>Introduction .....</b>	<b>6</b>
<b>3</b>	<b>General mode of operation .....</b>	<b>7</b>
<b>4</b>	<b>Recording architecture types .....</b>	<b>9</b>
4.1	Basic architecture types .....	9
4.1.1	All-in-one Basic .....	9
4.1.2	Multi-Server Recording .....	10
4.2	Architecture types for failover concepts .....	10
4.2.1	All-in-one Failover .....	11
4.2.2	Multi-Server Failover .....	12
4.3	Architecture types for parallel recording.....	14
4.3.1	All-in-one Parallel Recording.....	14
4.3.2	Multi-Server Parallel Recording .....	15
4.4	Architecture types for import .....	15
<b>5</b>	<b>Redundancy options.....</b>	<b>16</b>
<b>6</b>	<b>Synchronization options .....</b>	<b>18</b>
6.1	Synchronizing recording control .....	18
6.2	Synchronization of system storage .....	20
6.2.1	Create synchronization configuration.....	21
6.2.2	Delete synchronization configuration .....	21
<b>7</b>	<b>Supported system architectures .....</b>	<b>22</b>
7.1	Single-server system.....	23
7.2	Multi-server systems .....	23
7.2.1	Exemplary installation .....	24
7.3	Single-core system.....	27
7.4	Multi-server system with multi-cores .....	28
7.5	Redundant database instances.....	30
7.6	Redundant recording components .....	31
<b>8</b>	<b>Recommended system architectures .....</b>	<b>32</b>
8.1	Default architecture 1 .....	32
8.2	Default architecture 2 .....	33
8.3	Default architecture 3 .....	34
8.4	Default architecture 3 a .....	35
8.5	Default architecture 3 b .....	36
8.6	Default architecture 4 .....	38
8.7	Default architecture 4 a .....	39
<b>9</b>	<b>Definition of the terms .....</b>	<b>40</b>

9.1	System, general .....	40
9.2	Servers, types, and functions .....	41
9.3	Recording types .....	43
9.3.1	Recording .....	44
9.3.1.1	EVOIPneo .....	44
9.3.1.2	EVOLUTIONneo .....	44
9.4	Drive categories .....	44
	<b>List of figures .....</b>	<b>46</b>
	<b>List of tables .....</b>	<b>48</b>
	<b>Glossary .....</b>	<b>49</b>

## 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.

The software programs and the manuals of ASC are protected by copyright law. All rights on the manuals are reserved including the rights of reproduction and multiplication of any kind, be it photo mechanical, typographical or on digital data media. This also applies to translations. Copying the manuals, completely or in parts, is only allowed with written authorization 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 "Supported system architectures"](#), p. 22.

Within the different system architectures, different recording architectures can be used, see [chapter "Recording architecture types"](#), 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. 16.

Basically, the **recording system** 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

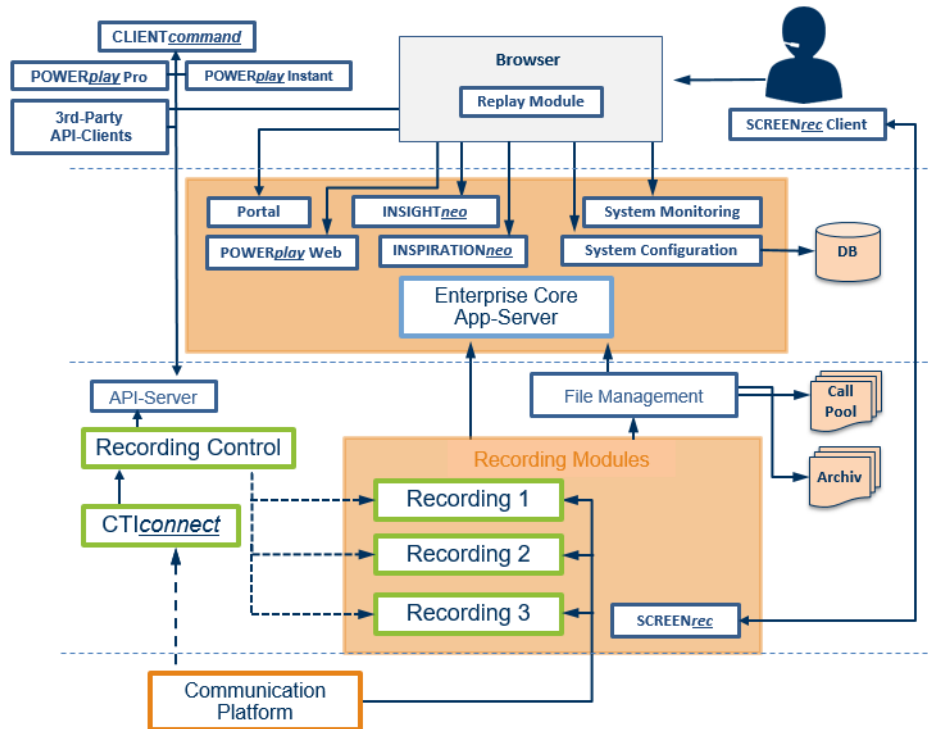


Fig. 1: Basic structure of the recording system

The following components may be installed on different servers:

- [App server](#)
- Recording components and [API server](#)
- Database

This results in several possible system architectures.



Information about which system architectures are possible in general can be found in [chapter "Supported system architectures"](#), p. 22.

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

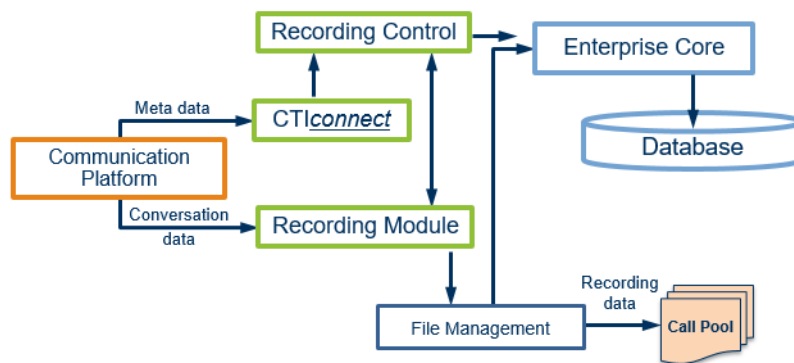


Fig. 2: Data stream in the recording process

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.
- CTIconnect (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 some recording architectures, you can additionally activate the individual recording components 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 architecture types have been stored in the system see [chapter "Recording architecture types", p. 9](#).



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



## 4 Recording architecture types

The following color code is used to illustrate the recording architectures:

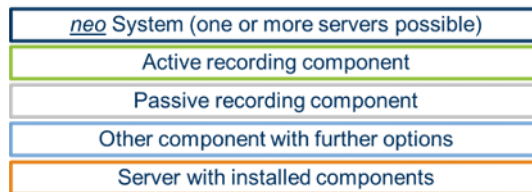


Fig. 3: Color code recording architectures:

The neo recording system supports the following recording architecture types:

### 4.1 Basic architecture types

#### 4.1.1 All-in-one Basic

With an 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.

A recording system may include several independent all-in-one architectures.

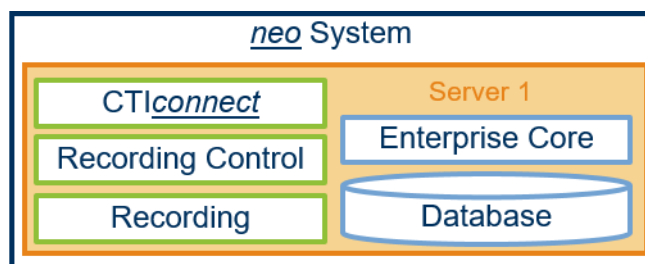


Fig. 4: Architecture type All-in-one Basic

#### All-in-one Basic Recording

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.

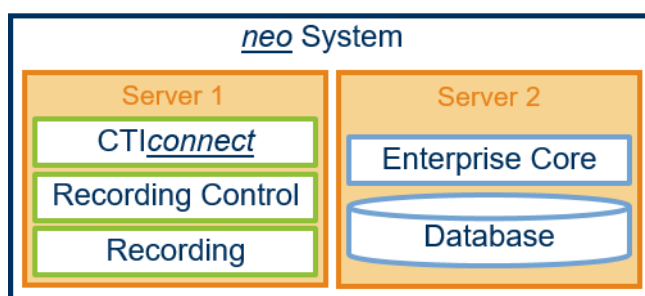


Fig. 5: Architecture type All-in-one Basic with 2 servers

#### 4.1.2 Multi-Server Recording

An architecture of this type allows distributing the load on several [recording servers](#). The pool of recording servers can contain any number of recording servers. The other recording components (Recording Control service and CTIconnect) must not be set up redundantly.

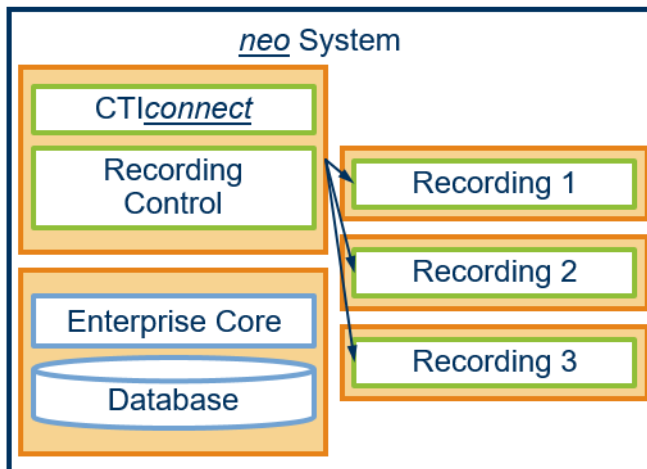


Fig. 6: Architecture type Multi-server recording with redundant recording servers

Besides the redundant recording servers, the recording control components CTIconnect and Recording Control may be set up redundantly in multi-server systems.

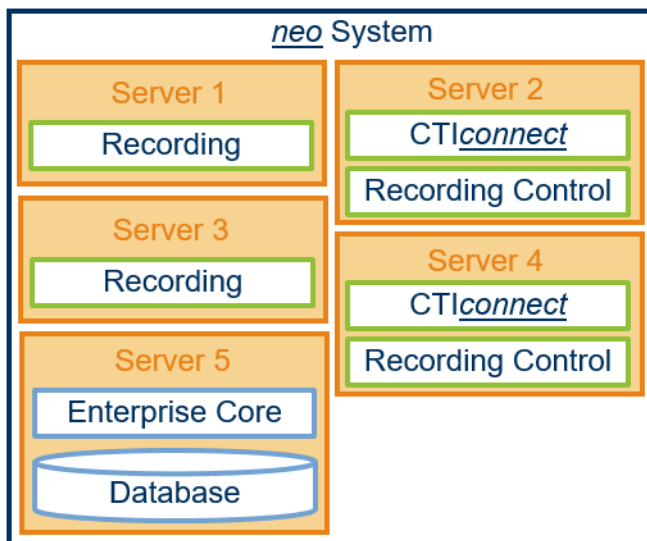


Fig. 7: Architecture type Multi-server recording with redundant recording control

#### 4.2 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.



#### **DANGER!**

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.



### DANGER!

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

#### 4.2.1

##### 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.

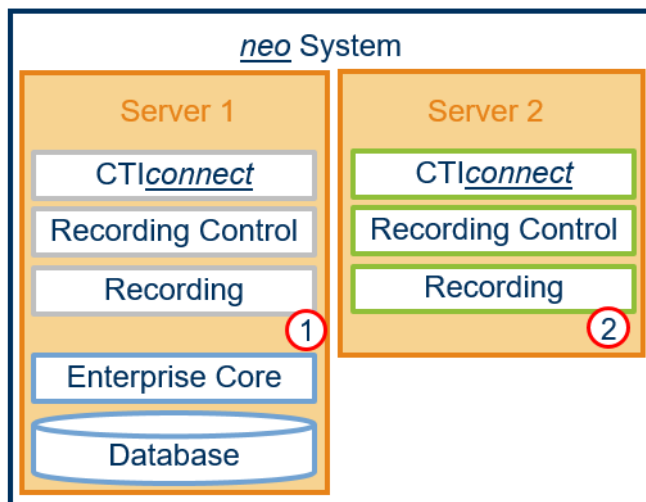


Fig. 8: Architecture type All-in-one Failover

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*.

#### 4.2.2 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 CTIconnect 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.

When using the architecture type *Multi-Server Failover*, the following architecture scenarios can be implemented among others:

##### 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.

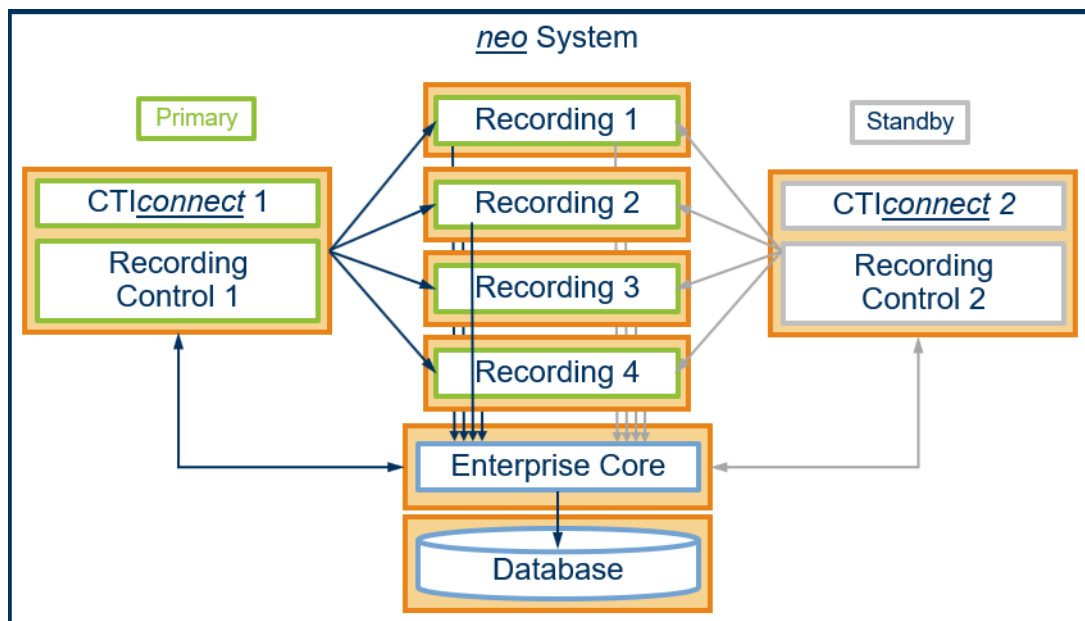


Fig. 9: Architecture type Multi-Server Failover with single pool of recording servers

### Multi-server failover with all redundancy options

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.

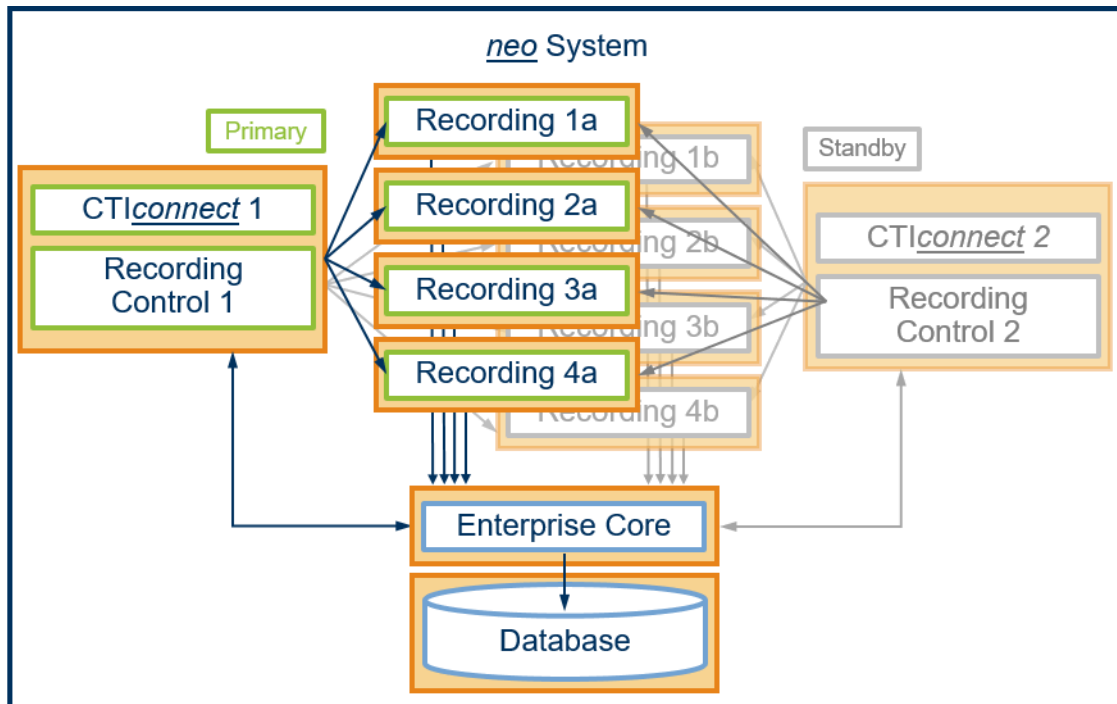


Fig. 10: Architecture type multi-server failover with all redundancy options

### 4.3 Architecture types for parallel recording

The purpose of a parallel recording is to avoid a loss of recordings in case a recording component fails. To this end, two recording trunks are configured which are active simultaneously so that each recording is captured twice. If one recording trunk fails, the recording is captured at least by the other recording trunk.

For parallel recording systems, there are different configuration possibilities:

- When the parallel recording servers have been installed and configured independently of each other, the recording servers record separately. In the event of an error, none of the servers can take over the tasks of another server.
- If the parallel recording servers have been integrated in a system architecture, the recording servers can take over the tasks of another server and the two servers can replace each other. See [chapter "Synchronizing recording control", p. 18](#)

#### 4.3.1 All-in-one Parallel Recording

This recording architecture is also known as Active-Active 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.

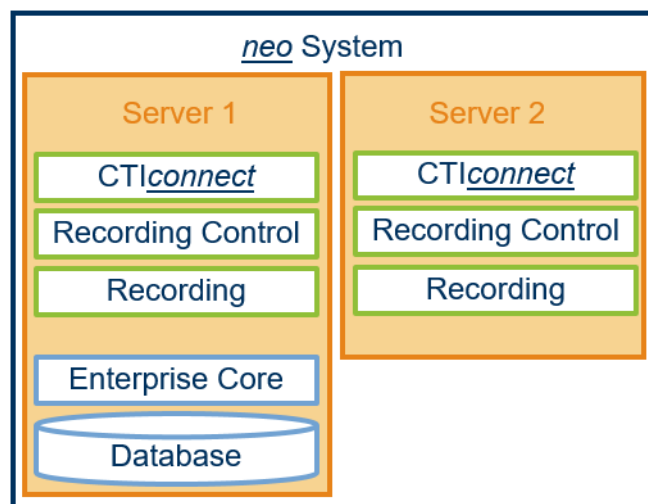


Fig. 11: Architecture type All-in-one Parallel Recording



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

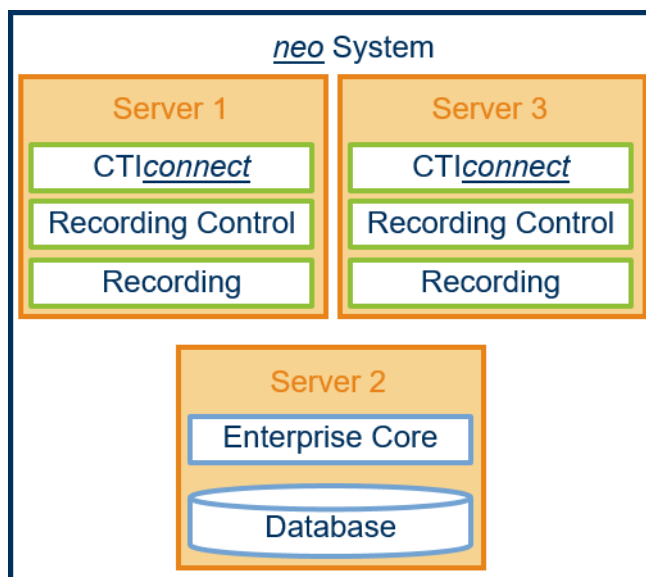


Fig. 12: Architecture type All-in-one Parallel Recording with 3 servers

#### 4.3.2 Multi-Server Parallel Recording

In an architecture of this type, the recording components of the two recording trunks are distributed on several servers. Every recording trunk has its own pool of recording servers.

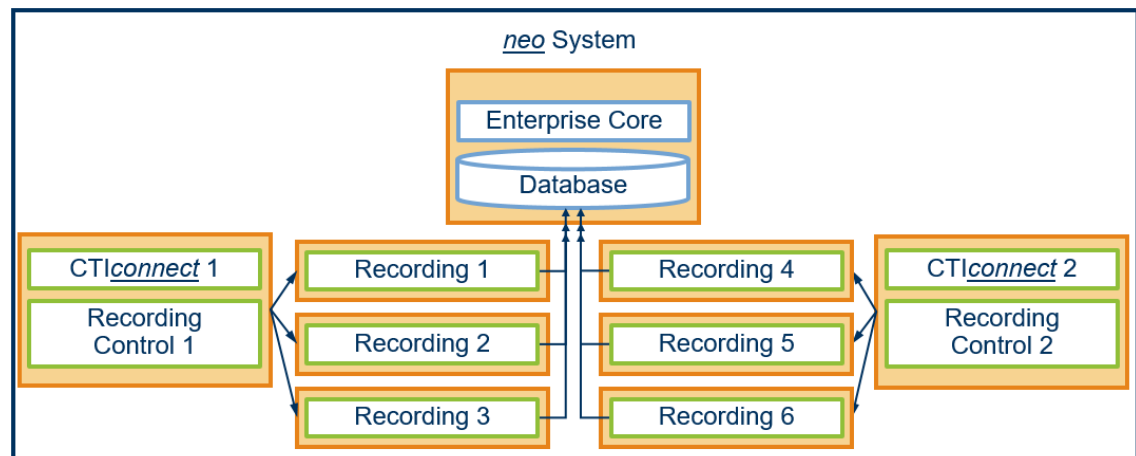


Fig. 13: Architecture type Multi-Server Parallel Recording

#### 4.4 Architecture types for import

The architecture type *Import only* offers the possibility to configure a simple architecture specifically for the 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 architecture type cannot be used for recording as it neither contains a [recording server](#) nor Recording Module. This architecture type exclusively consists of 1 recording component *RecordingControl*.



Theoretically, an import works with any other architecture type, too. Therefore, you can use an architecture for the import that you are already using for an integration.

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 [chapter "Multi-server system with multi-cores"](#), p. 28.
- Database  
See [chapter "Redundant database instances"](#), p. 30.
- Individual recording components  
See [chapter "Redundant recording components"](#), p. 31.
- Entire recording trunks (Recording modules service, CTIconnect service, and Recording Control service)  
See chapters:  
[chapter "Architecture types for failover concepts"](#), p. 10  
[chapter "Architecture types for parallel recording"](#), p. 14

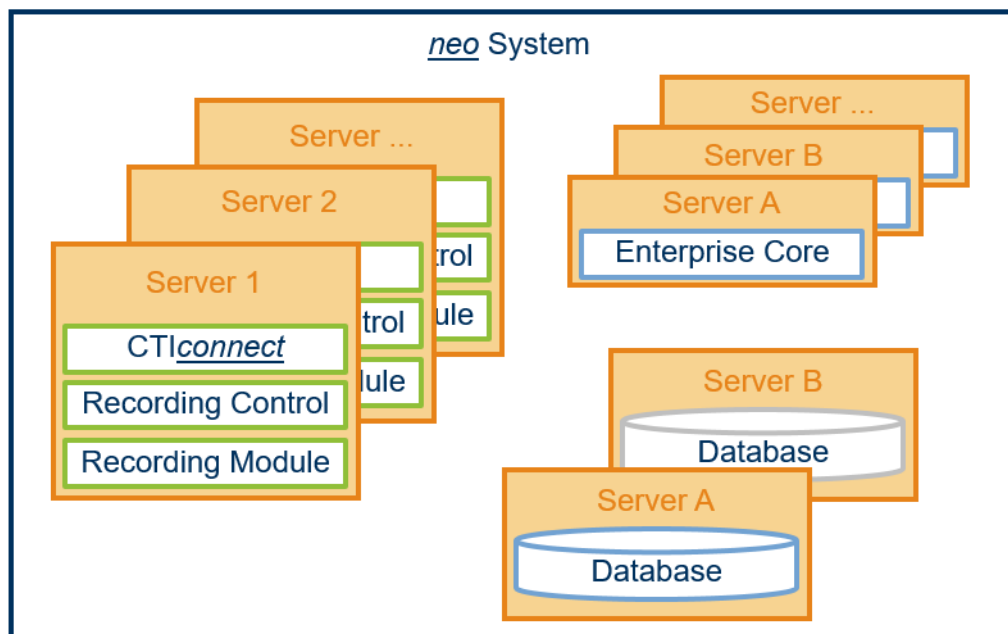


Fig. 14: Redundancy options



## Required number of servers:

Architecture	Redundancies			
	No redundancy	Recording redundancy	Enterprise core redundancy	Core redundancy and database redundancy
All-in-one basic	1 Starting with 1 server ✓	2 ✗	Starting with 3 servers ✓	Starting with 4 servers ✓
All-in-one failover	✗	3 Starting with 2 servers ✓	Starting with 3 servers ✓	3+ Starting with 4 servers ✓
All-in-one parallel recording	✗	4 Starting with 3 servers ✓	Starting with 3 servers ✓	4+ Starting with 4 servers ✓
Multi-server recording	Starting with 2 servers ✓	✗	Starting with 3 servers ✓	Starting with 4 servers ✓
Multi-server failover	✗	Starting with 4 servers ✓	Starting with 4 servers ✓	Starting with 6 servers ✓
Multi-server parallel recording	✗	Starting with 5 servers ✓	Starting with 5 servers ✓	Starting with 6 servers ✓

 Default scenarios

 Advanced default scenarios 3 and 4 requires the support of Solutions Consulting

## 6

## Synchronization options

There 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

## Synchronizing recording control

## Recording Control services

In parallel recording servers which have been installed and configured in the same system architecture, you can configure the synchronization of recording control.

**DANGER!**

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.

---

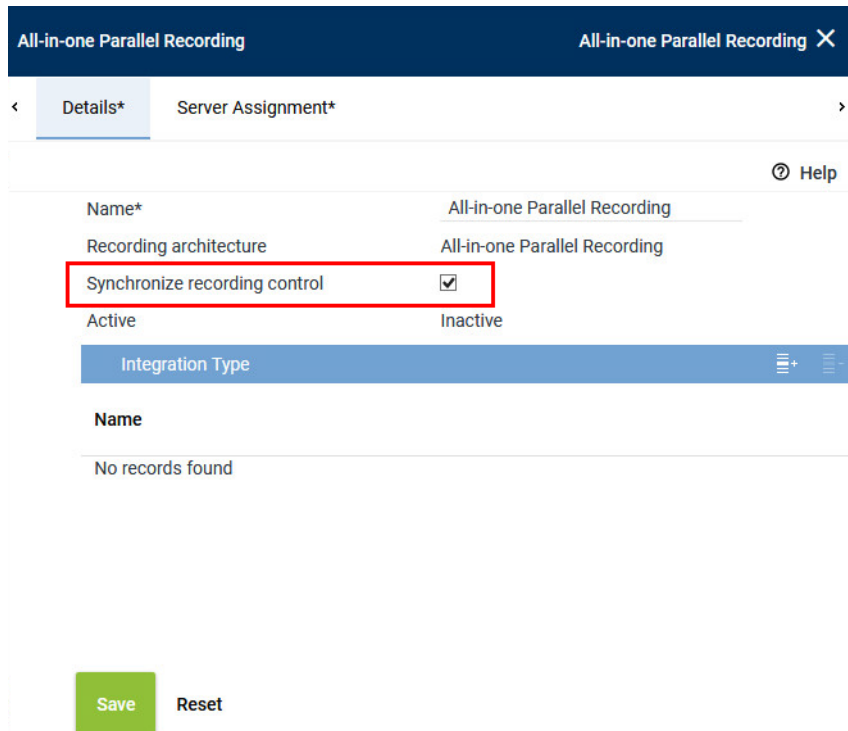
If recording control is supposed to take place by means of external applications such as *CLIENTcommand*, *PHONEapp*, or *SCREENrec* scan Editor, a synchronization of the Recording Control services of the parallel recording servers must be set up.

Primarily, recording control is carried out by the 1st Recording Control service. The Recording Control service guarantees that the conversations are recorded by both recording servers.

If the 1st Recording Control service fails, the 2nd Recording Control service takes over the task of recording control for both recording servers, both of which will record the conversations then.

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.



The screenshot shows a configuration window titled 'All-in-one Parallel Recording'. It has two tabs: 'Details\*' and 'Server Assignment\*'. The 'Details\*' tab is selected. Inside, there's a 'Name\*' field with 'All-in-one Parallel Recording' and a 'Recording architecture' field with 'All-in-one Parallel Recording'. A checkbox labeled 'Synchronize recording control' is checked and highlighted with a red rectangle. Below it, the 'Active' status is 'Inactive'. At the bottom, there are 'Save' and 'Reset' buttons.

Fig. 15: Synchronize recording control

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

**If you subsequently activate or deactivate this synchronization options, you have to carry out the following configuration steps again before the changes take effect:**

1. Set the requested state of the recording control:
  - ☒ = *recording control is synchronized*
  - ☐ = *recording control is not synchronized*
2. Deactivate the integration.
3. Deactivate the recording architecture.
4. Check 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 may be in a conflict status.**

**Therefore, you have to carry out the following steps again:**

6. Deactivate the recording architecture again.
  7. Check that the following services have been stopped.
  8. Activate the recording architecture again.
  9. Activate the integration.
- ⇒ Now, the changes have been applied.

## 6.2 Synchronization of system storage

In recording architectures with 2 system storages, you can configure a synchronization for comparing the recordings.

A synchronization configuration is always created for 2 system storages. All recordings which are added to one system storage are copied to the other system storage, too, and vice versa. That way, all recordings of both system storages are available on the 2 system storages simultaneously. If one of the two system storages fails, you can thus access the recordings of the failed system storage via the other system storage.

Synchronization of system storage 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. 16: Menu item Manage synchronization configuration

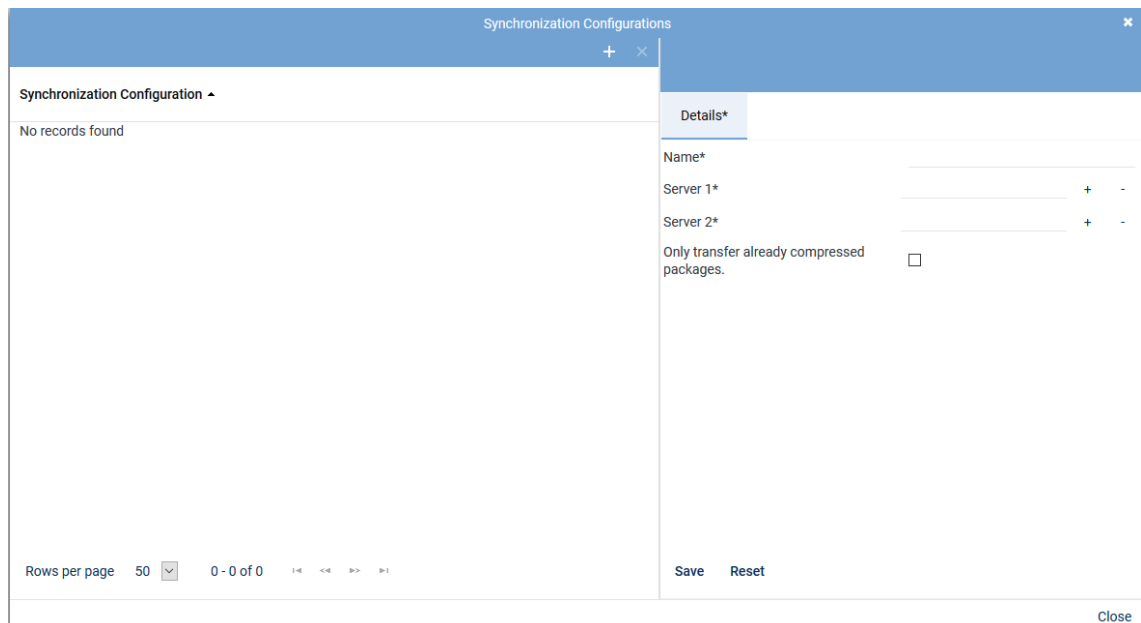




Fig. 17: Configure synchronization configurations

The following options are available:


	<b>Create</b>	Creates a new synchronization configuration (see <a href="#">chapter "Create synchronization configuration", p. 21</a> ).
	<b>Delete</b>	Deletes the selected synchronization configuration (see <a href="#">chapter "Delete synchronization configuration", p. 21</a> ).

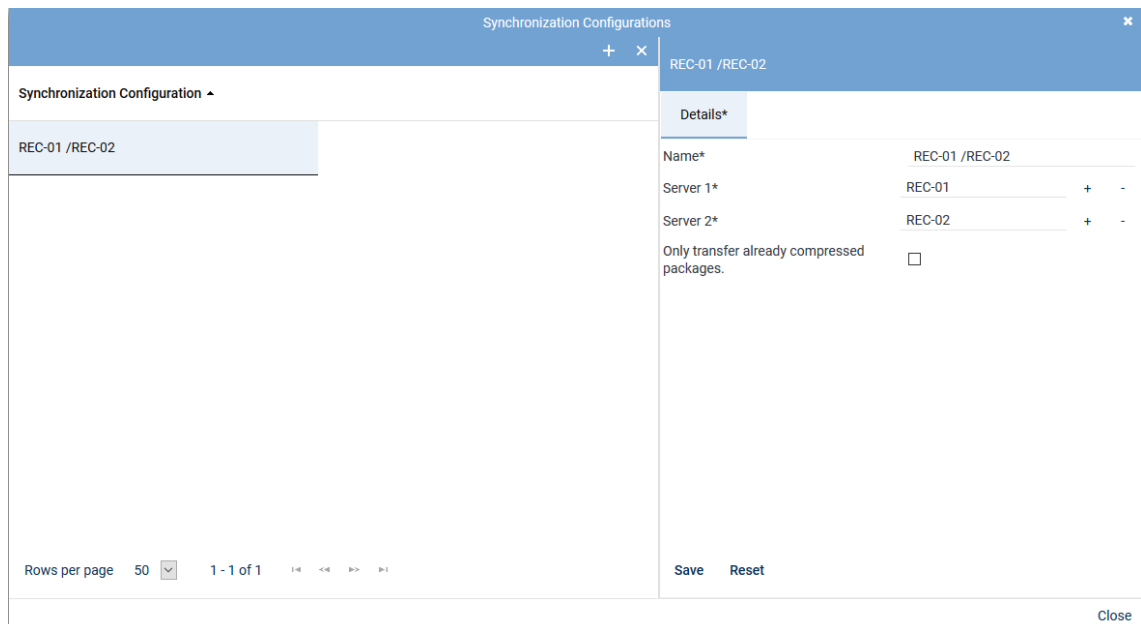
A synchronization configuration becomes active upon saving it and continues running until it is deleted. During this period both system storages are regularly checked for new content and synchronized.



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

### 6.2.1 Create synchronization configuration

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





The screenshot shows a window titled "Synchronization Configurations" with a toolbar containing a "+" (Create) and a "x" (Close) icon. The main area is divided into two panes. The left pane shows a list of configurations with "REC-01 / REC-02" selected. The right pane, titled "Details\*", contains the following fields:

- Name\***: REC-01 / REC-02
- Server 1\***: REC-01 (with a "+" button to the right)
- Server 2\***: REC-02 (with a "+" button to the right)
- Only transfer already compressed packages.**: ☐

At the bottom of the right pane are "Save" and "Reset" buttons. The bottom of the window has a "Close" button.


Fig. 18: Create synchronization configuration

- Complete all fields for the new synchronization configuration:

<b>Name</b>	Enter a name for the synchronization configuration.
<b>Server 1 / Server 2</b>	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.
<b>Only transfer already compressed packages</b>	Select whether data which has not yet been compressed is supposed to be transferred, too. <input checked="" type="checkbox"/> = Uncompressed data is transferred, too. <input type="checkbox"/> = Only compressed data is transferred.  <b>NOTICE!</b> This option is not available until you have entered and saved the two servers.

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

### 6.2.2 Delete synchronization configuration

- In the window *Administrate synchronization configurations*, select the synchronization configuration you would like to delete.
- 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.

## Supported 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:

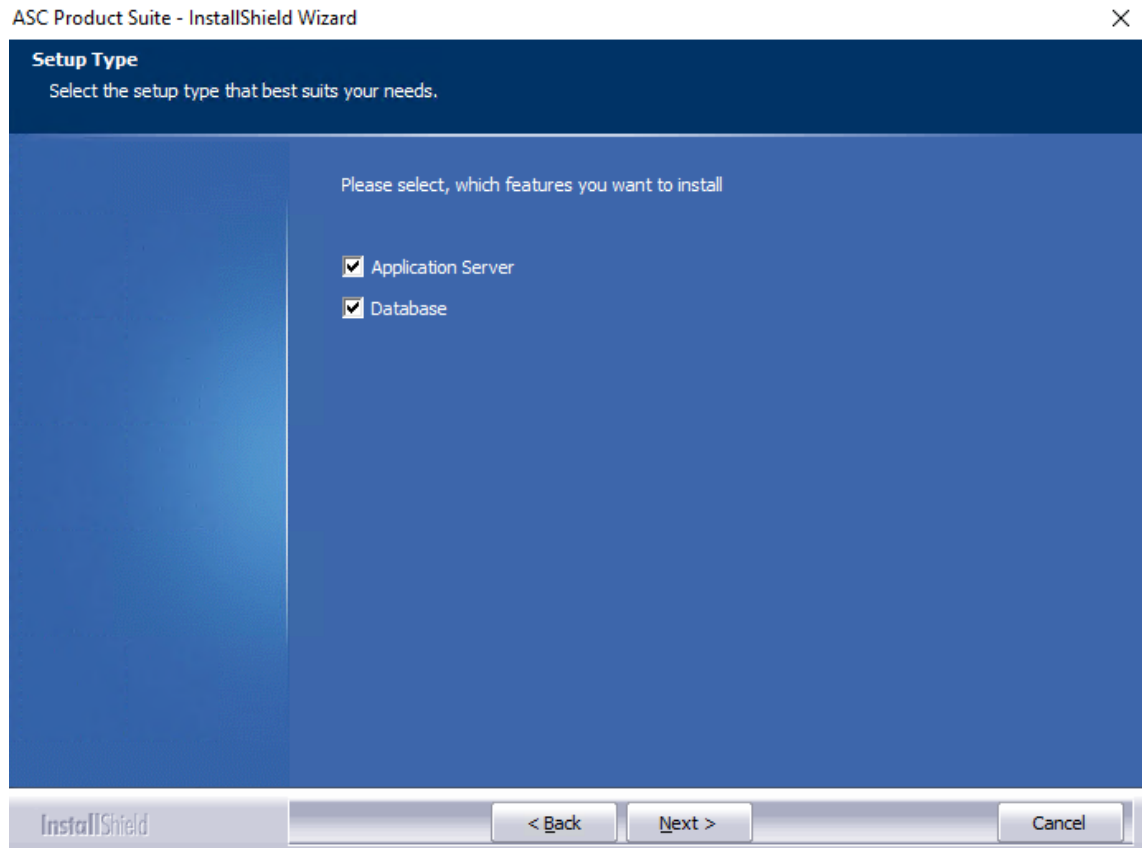


Fig. 19: Selection of optional software components

- **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).

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 [chapter "Multi-server system with multi-cores", p. 28](#).

- [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 PostgreSQL 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. 30](#).

## 7.1

### 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.

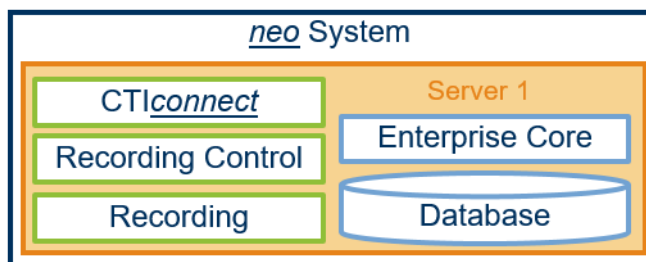


Fig. 20: Single-server system

## 7.2

### Multi-server systems

In multi-server systems, the *neo* software is distributed on several servers and can thus be installed redundantly, too. There are different possibilities of distribution and redundancies.

### 7.2.1 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 along with the database on a second server.

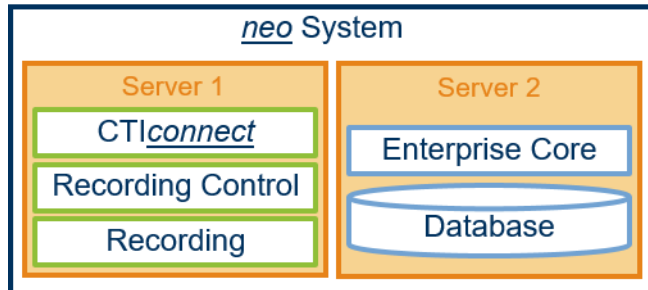


Fig. 21: Multi-server system with 2 servers and all-in-one recording

#### Multi-server system with 3 servers with multi-server recording, separate Enterprise Core and internal database

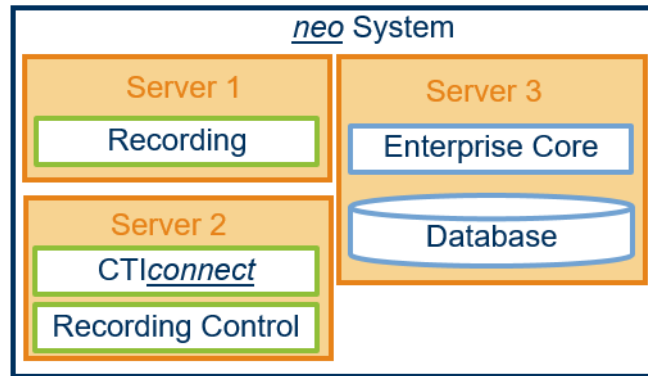


Fig. 22: Multi-server system with 3 servers, multi-server recording and separate Enterprise Core with database

#### Multi-server system with 3 servers with all-in-one recording, separate Enterprise Core and external database

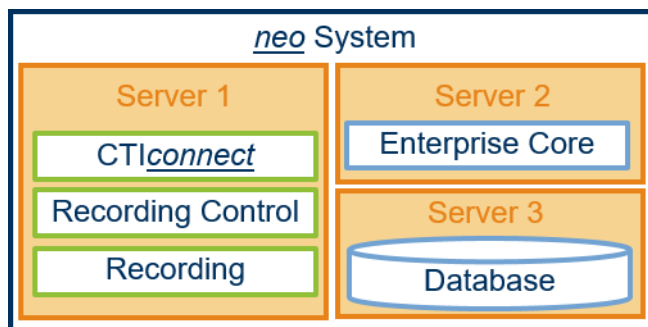


Fig. 23: Multi-server system with all-in-one recording, separate Enterprise Core and separate database



**Multi-server system with 4 servers with parallel all-in-one recording, separate Enterprise Core and external database with failover option each**

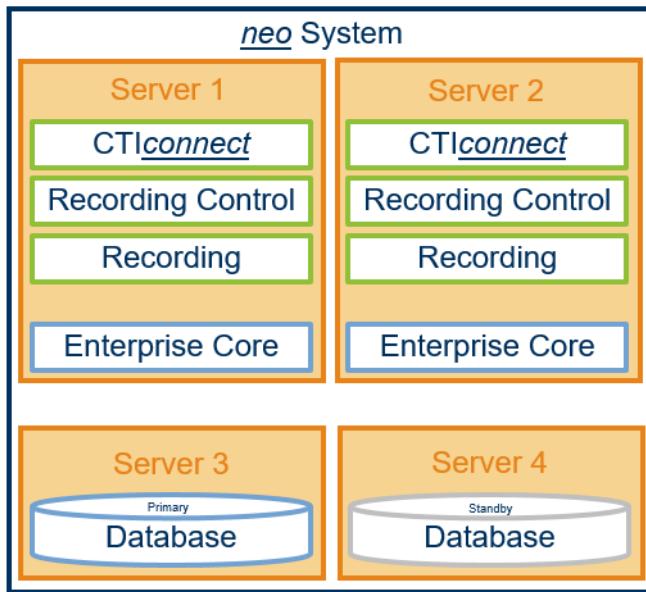


Fig. 24: Multi-server system with All-in-one Parallel Recording and separate failover database

For this system architecture, you have to create a reciprocal synchronization job on server 1 and server 2. All recordings which are added to one system storage are copied to the other system storage, too, and vice versa. That way, all recordings of both system storages are available on the 2 system storages simultaneously. If one of the two system storages fails, you can thus access the recordings of the failed system storage via the other system storage.



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

Additionally, individual recording components can be installed 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**

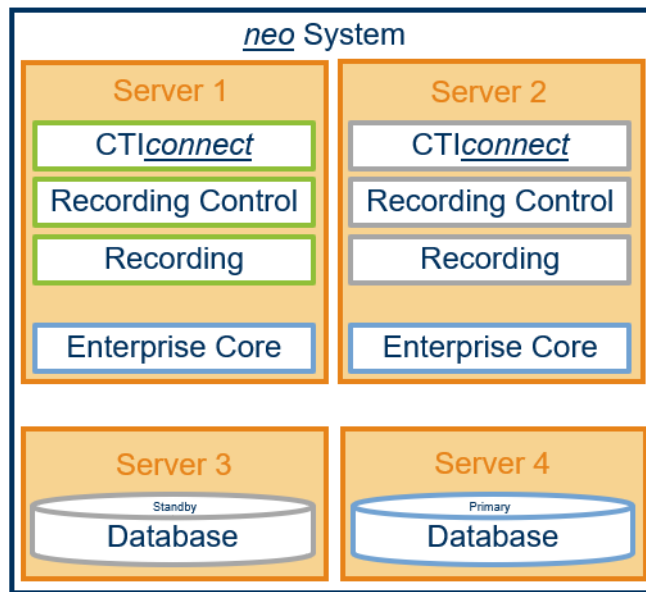


Fig. 25: Multi-server system with 4 servers with All-in-one Failover Recording, redundant Enterprise Core and external redundant database

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.

The redundant design of the Enterprise Core and the database guarantees that the recorded conversations are stored in the system at any moment and are immediately retrievable. In case of a failure, 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 5 servers with parallel multi-server recording, separate Enterprise Core and internal database**

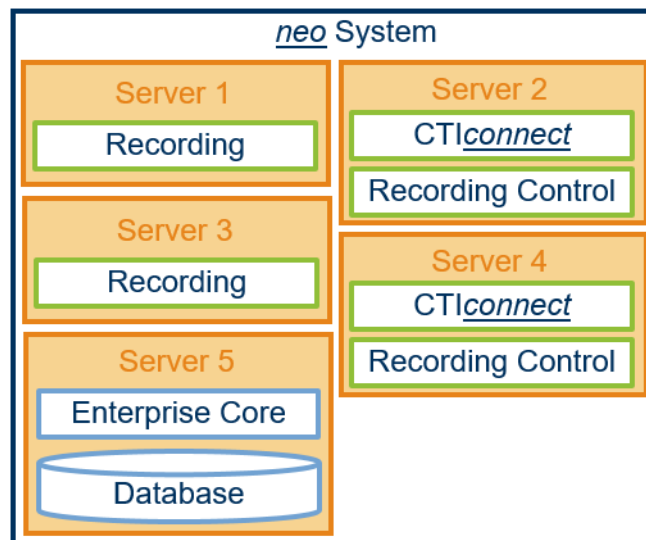


Fig. 26: Multi-server system with 5 servers, multi-server parallel recording and separate Enterprise Core with database

The recording components on different servers can record different recording trunks or may be configured as a redundancy to record in parallel.

## 7.3

**Single-core system**

In this solution, there is only one Enterprise Core which may be installed on the same server as all other components. As an alternative, the recording modules and the database may be installed separately on other servers and set up redundantly, too. Single-core refers exclusively to an Enterprise Core.

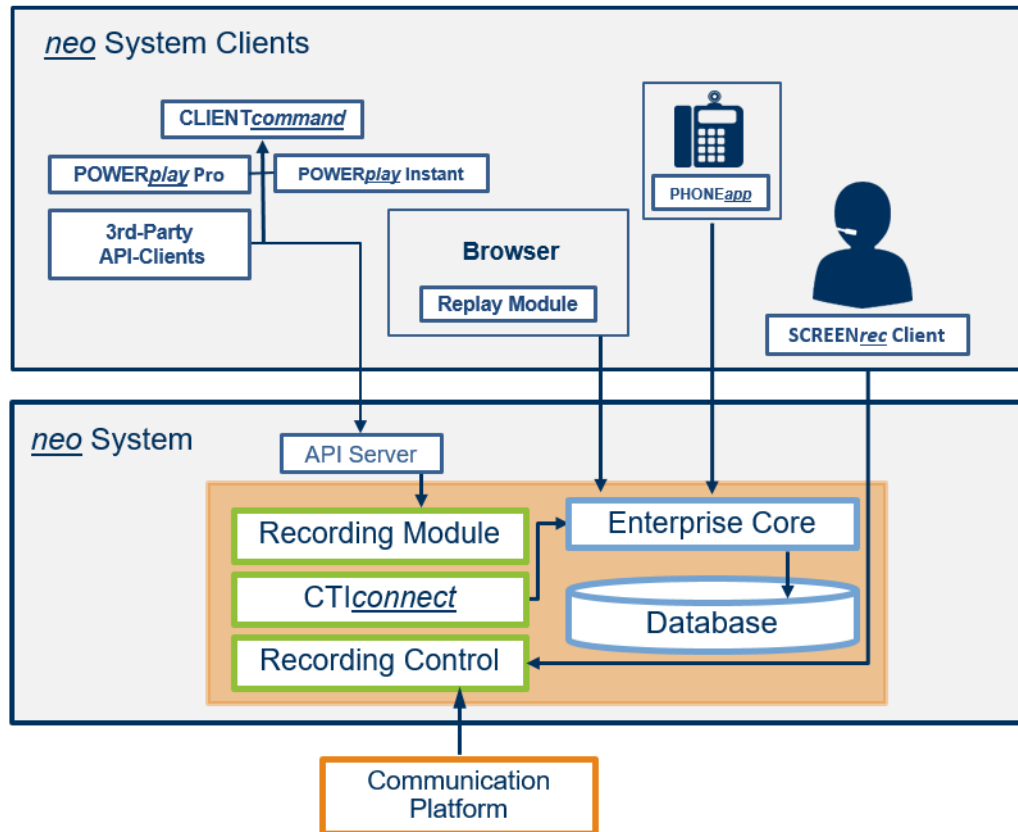


Fig. 27: Principle single-server system with single core

#### 7.4 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 distributed load sharing or to secure the recording in the event of a failure of the application server ([app server](#)), you can set up several application servers in a farm. In this server farm, the system load is distributed automatically among the different application servers. If an application server fails, the other application servers share all tasks among each other. All application servers use the same database.

During the installation of the recording software you can configure which application servers are available in your recording system and should be used 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.

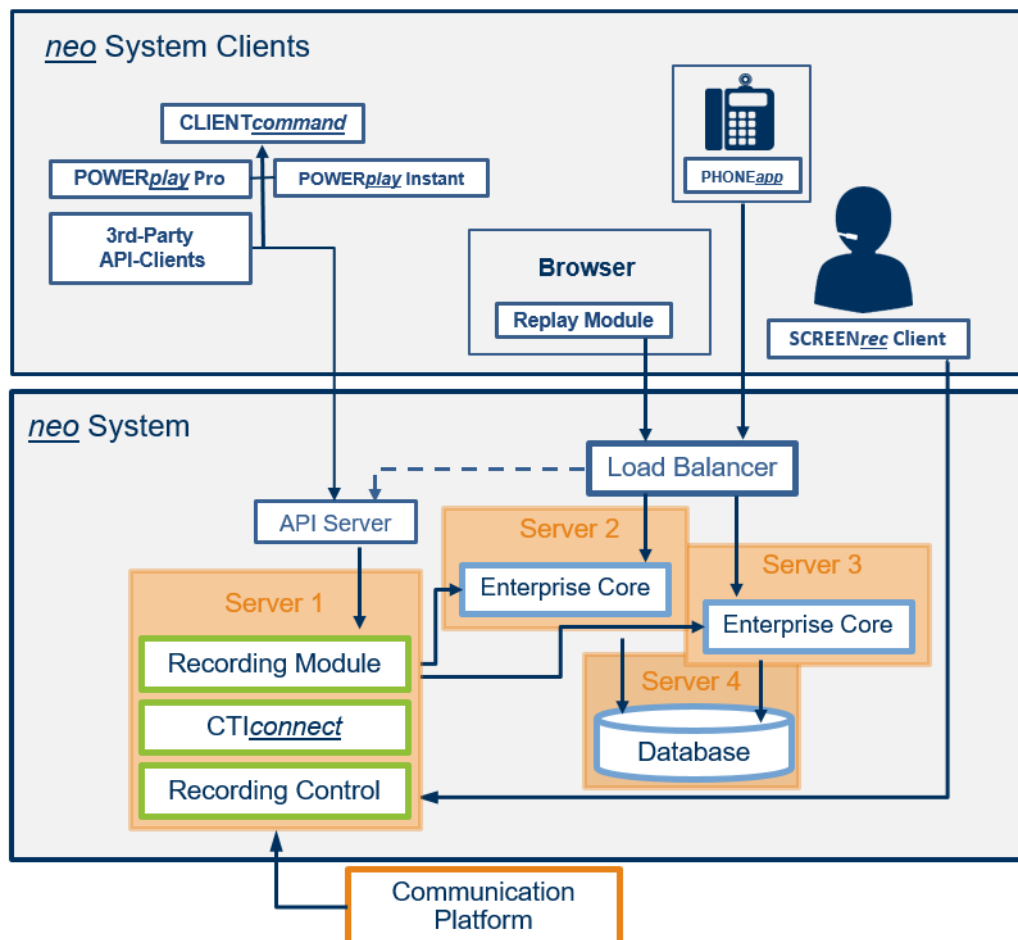


Fig. 28: Principle Multi-server system with multi-cores

When using a multi-core system, ASC recommends to set up the following components redundantly to increase recording safety and protect the recordings against unauthorized access:

- *Recording components*
- *Databases*

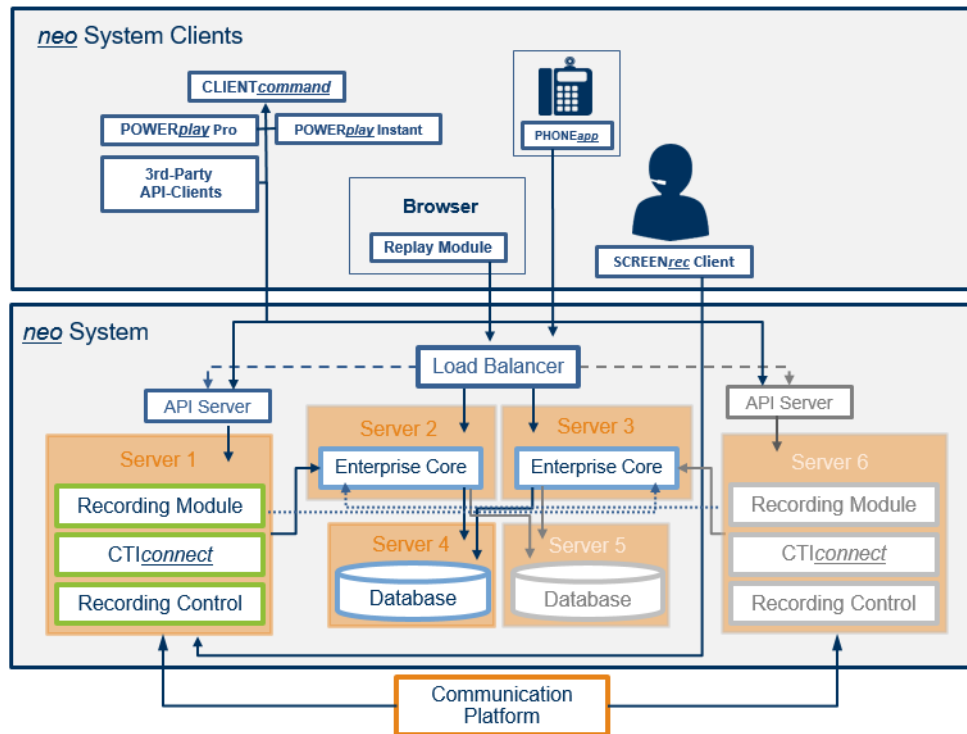


Fig. 29: Principle Multi-server system with multi-cores with all redundant components

In addition, configure a data transfer to all application servers on all **recording servers**. This ensures that all recordings are available as long as at least one application server and one data-base operate correctly.



You can configure the function *Data transfer* in the Servers module of the application System Configuration.



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

### 7.5 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 a PostgreSQL database, you can install a second PostgreSQL database instance on basis of the ASC 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.

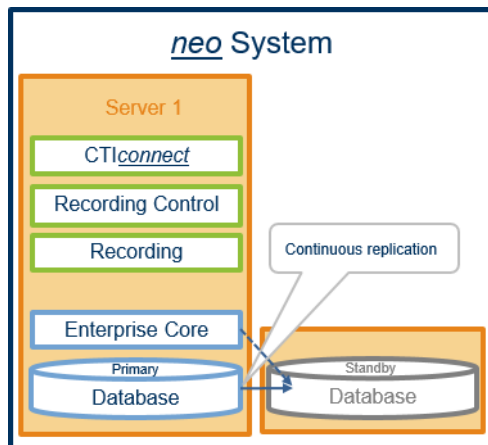


Fig. 30: Redundancy solution for PostgreSQL databases



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

#### 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.

**NOTICE!** The network latency between Enterprise Core and database must be  $\leq 10$  milliseconds.

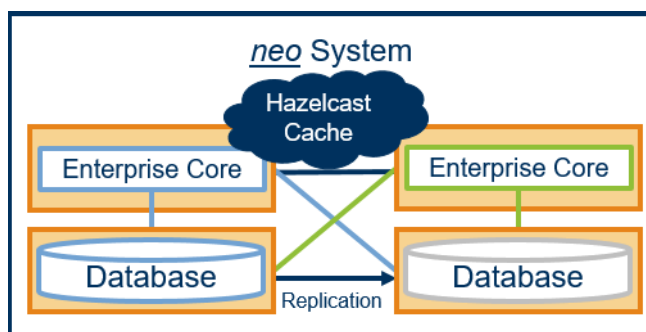


Fig. 31: Failover operation for 2 Enterprise Cores and redundant database

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*.

## 7.6

### Redundant recording components

To secure the recording in the event of a failure of a [recording server](#) or of a recording component, the application System Configuration enables you to set up different recording architectures where selected or all recording components have been setup redundantly.

Examples:

#### Multi-server recording architecture with redundant recording modules

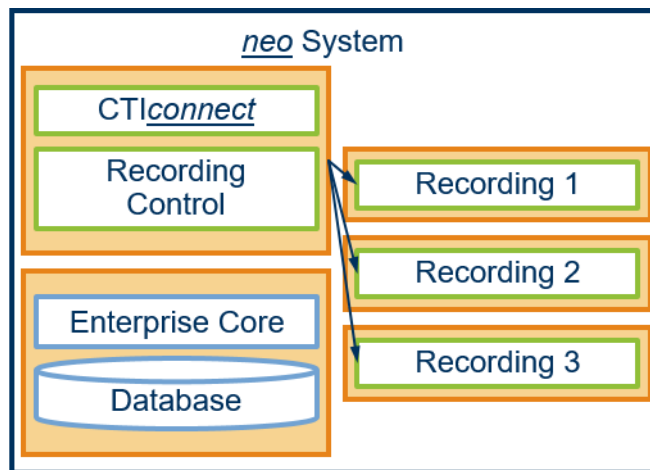


Fig. 32: Recording architecture with redundant recording modules

#### Multi-server recording architecture with redundant recording modules and recording control modules

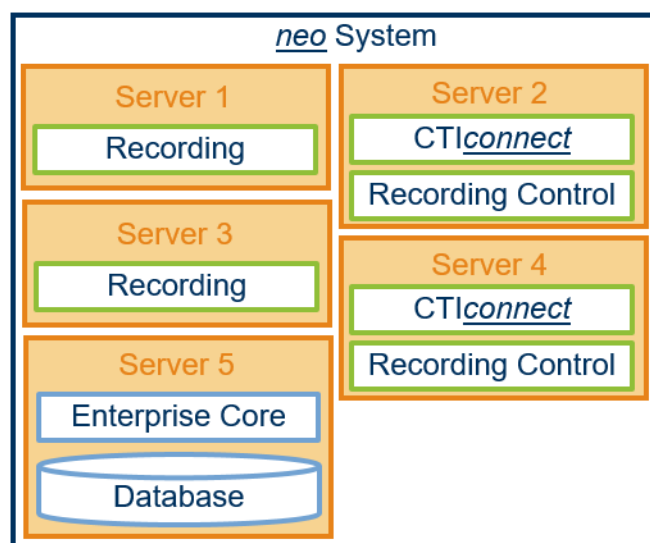


Fig. 33: Recording architecture with redundant recording modules and recording control modules



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

## 8 Recommended system architectures

## 8.1 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.

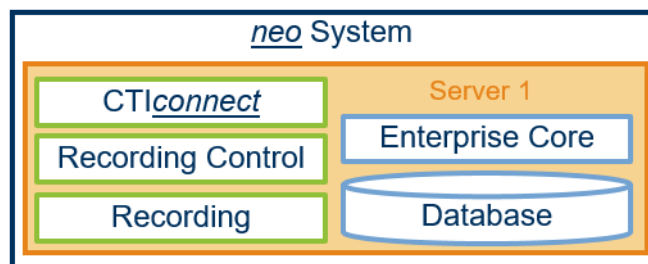


Fig. 34: Single-server system



## 8.2

## 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. The Enterprise Core and a database are installed on a second server to increase performance.

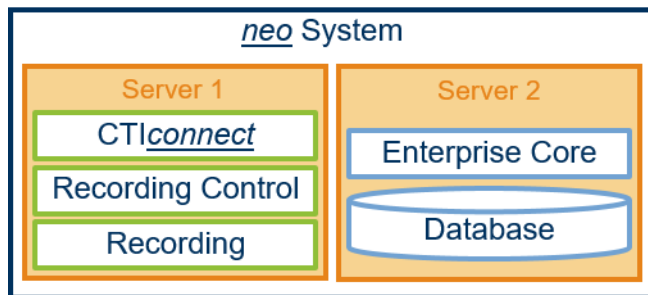


Fig. 35: System with 2 servers and 1 all-in-one recording architecture

Recordings can be replayed either on server 1 or on server 2:

1. To replay the recordings on server 1, a *replay server* must be configured there.
2. To replay the recordings on server 2, a *data transfer* from server 1 to server 2 must be configured. Data transfer can be configured for replay only or for data storage purposes as well.



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

## 8.3

## 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.

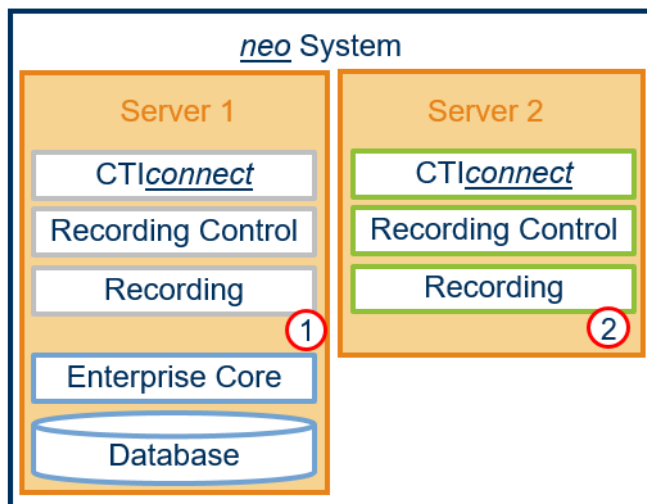


Fig. 36: System with 2 servers and All-in-one Failover

ASC recommends to set up a data transfer from server 2 to server 1 as server 1 will contain all recordings even if server 2 fails. If data is stored on a [NAS](#), a data transfer to the servers is not necessary.



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*.

## 8.4

## Default architecture 3 a

**All-in-one Failover recording architecture with 4 servers**

Default architecture 3 a (previously 3+) 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.

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

The redundant design of the Enterprise Core and the database guarantees that the recorded conversations are stored in the system at any moment and are immediately retrievable. 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.

**NOTICE!** The network latency between Enterprise Core and database must be  $\leq 10$  milliseconds.

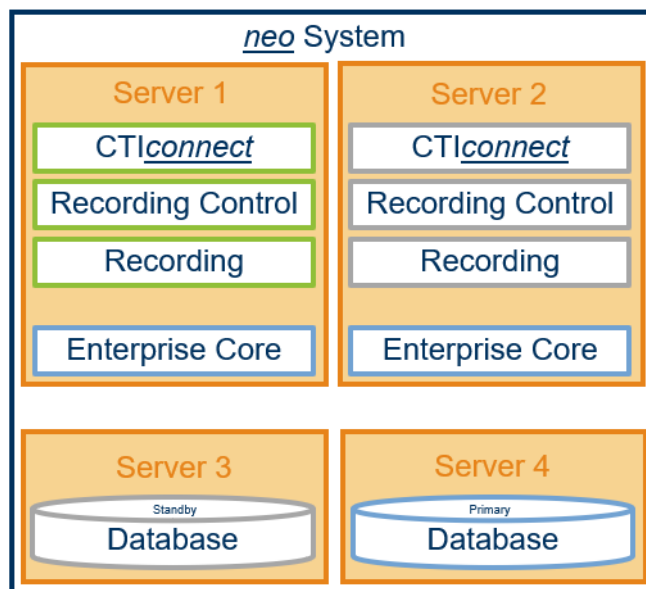


Fig. 37: Multi-server system with 4 servers, All-in-one Failover Recording, redundant Enterprise Core and external redundant database



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

## 8.5 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.

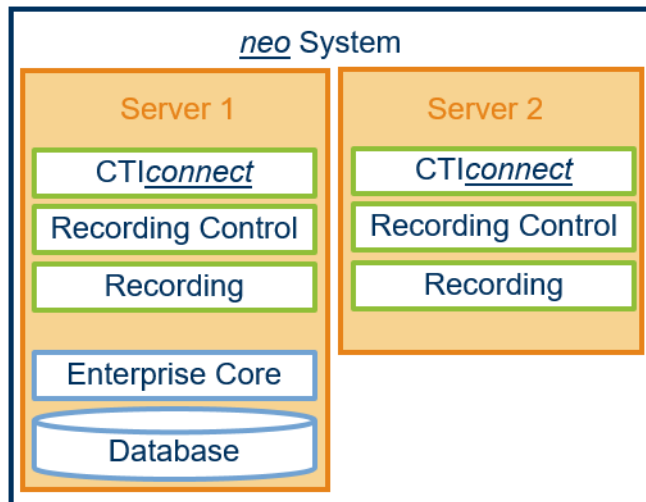


Fig. 38: System with 2 servers and All-in-one Parallel Recording



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

## Default architecture 4

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

Default architecture 4 consists of a system with 3 servers:

- 2 servers 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 separately as All-in-one Parallel Recording on different servers. 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. In addition, the Enterprise Core is installed along with the database on a third server. Neither of them is redundant but can be expanded accordingly.

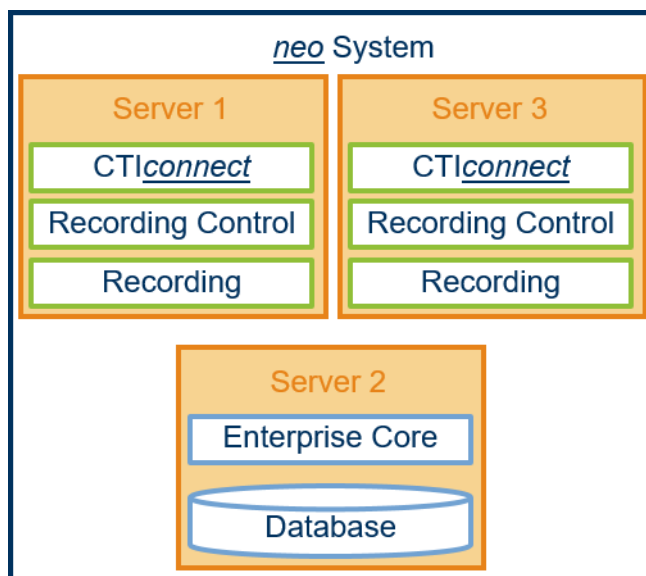


Fig. 39: System with 3 servers and All-in-one parallel Recording

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

If data is stored on a [NAS](#), a data transfer to the servers is not necessary.



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

## 8.7

## 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.

**NOTICE!** The network latency between Enterprise Core and database must be  $\leq 10$  milliseconds.

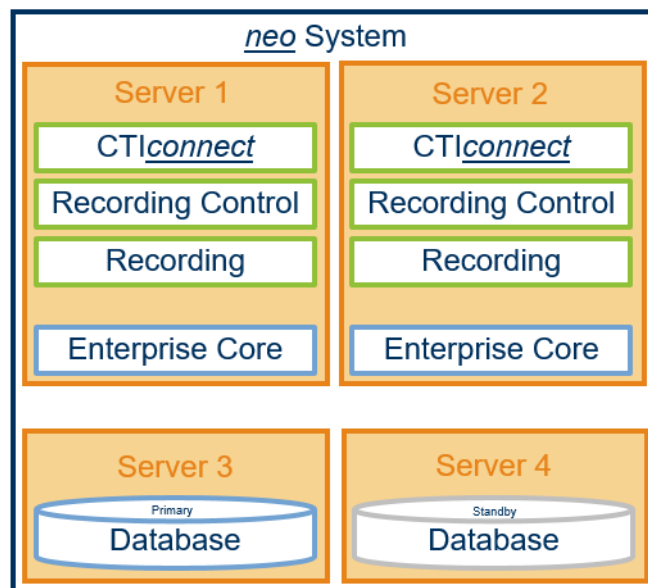


Fig. 40: System with 4 servers and All-in-one Parallel Recording



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

## 9 Definition of the terms

### 9.1 System, general

#### Recording architecture

Composition of all required recording components.

- Recording Control  
This service controls the recording according to the recording plan.
- CTIconnect (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. For the system provider, a tenant is created automatically, too. However, the system provider cannot be considered a 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 that system providers and tenants have.

- 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.

## 9.2 Servers, types, and functions

### Server

The term “server” is not necessarily restricted to hardware but may extend to a service or a function which (also in combination with other services or functions) has been installed and activated on hardware.

Example:

The entire recording software of ASC including the [app server](#) components have been installed on a server (hardware). 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](#). Via the user interface of the application System Configuration, further function can be activated on the server. If e. g. the function “Replay” is activated, then the server serves as [replay server](#), too. The server (hardware) thus serves as application server, recording server, and replay server at the same time.

### Application server

([App server](#))

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](#)).

*What makes a server an app server?:*

To do:

1. Activate the option *Application Server* during the installation of the ASC recording software.  
⇒ **App server** components are installed.

### API server

Server on which the **API** service runs.

### Recording server

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.

*What makes a server a recording server?:*

Precondition: ASC recording software has been installed

To do:

1. In the application System Configuration > Recording Architectures module > tab *Server Assignment* > use the server in a recording architecture

### Data storage server

*What makes a server a data storage server?:*

Precondition: ASC recording software has been installed

To do:

1. In the application System Configuration > Servers module > tab *Usage* > activate the function *Data storage*.
  2. The recordings from the call pools of the other servers are transferred to this server.  
(System Configuration > Servers module > tab *Usage* > activate function *Data storage* > *Transfer data for data storage* > enter data storage server as target server)
- ⇒ The server receives and saves the transferred recording data.  
In the Servers module > tab *Usage*, you can see the servers from which the data storage server receives data.

### Database server

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.

*What makes a server a database server?:*

To do for an internal PostgreSQL database:

1. Activate the option *Database* during the installation of the ASC recording software.  
⇒ PostgreSQL database is installed.

To do for an external database:

1. Installation of an ASC-supported database on a separate server.
2. Configuration of the connection to the database server during the installation of the ASC recording software.

### Replay server

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 are 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.

*What makes a server a replay server?:*

Precondition: ASC recording software has been installed

To do:

1. In the application System Configuration > Servers module > tab *Usage* > activate the function *Replay* and enter a name.
- ⇒ By means of the different replay applications of the recording system, the client computers 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*.

### 9.3

## Recording types









### Conversation

Umbrella term for the different types of communication which can be recorded. Used when it is not necessary to differentiate between the different conversation and media types.

### Conversation type

Type of communication, e. g. call, chat or [SMS](#).

The recorded data can 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, screen video
Calls with camera recording		Call	Audio, camera video
Calls with screen recording and camera recording		Call	Audio, screen video, camera video
SMS		<a href="#">SMS/SDS</a>	<a href="#">SMS/SDS</a> text
SDS		<a href="#">SMS/SDS</a>	<a href="#">SMS/SDS</a> text
Chat messages		Chat	Chat text

Tab. 1: Conversation types

When it comes to recordings, there is a difference between *call*, chat, text message, and *work item*.

- *Call*: conversation held via a phone. Any combination of call and [video recording](#) can be selected to be recorded.
- *Chat*:  
Conversation held via a chat platform.
- *Text message*:  
Conversation held via a Short Message System ([SMS](#)) or Short Data Service ([SDS](#)).
- *Work item*:  
Screen activity **without** reference to a call.

### Session

Recorded conversations are edited as sessions in INSPIRATION<sub>neo</sub>. A session is a section of a conversation in which one particular agent is active. Precondition for a session is that the recording planner has been activated in the System Configuration. 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 either con-

sist of only the agent's recorded phone call or additionally include the corresponding screen activity. The user has the possibility to assign agents sessions, to filter the sessions according to different criteria and thus to reduce the sessions to a restricted amount to be analyzed or evaluated.

### Difference between conversation and session

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

- 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.
- A session is a section of a conversation in which one particular 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 created at least one new session of its own. For each involved agent who is supposed to be recorded according to the recording plan a proper 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.

## 9.3.1 Recording

### 9.3.1.1 EVOIPneo

EVOIP<sup>neo</sup> is a voice documentation system with a powerful scalable platform. The system can be used as a stand-alone recorder or in a device group of several servers across locations.

EVOIP<sup>neo</sup> is a sophisticated recording technology developed to comply with legal regulations such as [MiFID II](#) or Dodd Frank. EVOIP<sup>neo</sup> offers [multi-channel recording](#) of voice, screen, video, and chat.

### 9.3.1.2 EVOLUTIONneo

EVOLUTION<sup>neo</sup> is an addition to supplement traditional telephony with an ASC recording system. Besides [VoIP](#) recording, EVOLUTION<sup>neo</sup> offers interfaces to all standard [TDM](#)-based PBXs by means of specifically developed recording cards in their own server cabinet designed for the purpose.

#### Definition of names:

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

## 9.4 Drive categories

During the configuration, the drives have to be mapped to a category. The category defines the functions available for the drive.

There are 5 different categories:

### 1. System storage

Drives which serve as system storage are mounted and configured during the installation. The system storage is exclusively used for the recording of conversations. The system storage can neither be used for archiving nor for import, export or as storage expansion.

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

You have the possibility to compress the recordings before saving them in the system storage.

## 2. **Storage expansion**

The storage expansion adds storage to the system storage. The capacity of a storage expansion has to exceed the capacity of the system storage by at least 10 %.

Any number of storage expansions can be set up per system storage. To activate a storage expansion for usage, you have to map the storage expansion to at least 1 tenant, though. This implies that you can use as many active (activated) storage expansions per system storage at maximum as there are tenants in the system.

All recordings of tenants who have been mapped to a storage expansion are copied to this storage expansion. Thus, the local availability of the recordings of those tenants is increased in case recordings are deleted from the system storage due to capacity reasons. The recordings of tenants who have not been mapped to a storage expansion are saved exclusively on the system storage.

You can map any number of tenants to the storage expansion.

The 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 only be used for archiving, import, and export.

You can configure any number of data drives.

## 4. **Database drive**

Exclusively the database is installed on the database drive. You cannot install any other software components on this drive. The database drive is created during the installation of the ASC software if you do not use an external database. A maximum of 1 database drive can be created per recording system.

## 5. **Application drive**

The ASC software is installed on the application drive. You can install other software components on this drive besides the ASC software, though. The application drive is created during the installation of the ASC software. The drive with the Windows installation is considered an application drive, too. Application drives can be used as the source drive for the import of conversations.



---

In virtual environments, you can only use network drives for archiving, import and export of data. Internal drives or [USB](#) drives are not supported as this may result in access and performance issues if the drive is unavailable.

---

## List of figures

Fig. 1	Basic structure of the recording system .....	7
Fig. 2	Data stream in the recording process .....	8
Fig. 3	Color code recording architectures: .....	9
Fig. 4	Architecture type All-in-one Basic .....	9
Fig. 5	Architecture type All-in-one Basic with 2 servers .....	9
Fig. 6	Architecture type Multi-server recording with redundant recording servers .....	10
Fig. 7	Architecture type Multi-server recording with redundant recording control .....	10
Fig. 8	Architecture type All-in-one Failover .....	11
Fig. 9	Architecture type Multi-Server Failover with single pool of recording servers.....	12
Fig. 10	Architecture type multi-server failover with all redundancy options .....	13
Fig. 11	Architecture type All-in-one Parallel Recording.....	14
Fig. 12	Architecture type All-in-one Parallel Recording with 3 servers .....	15
Fig. 13	Architecture type Multi-Server Parallel Recording .....	15
Fig. 14	Redundancy options .....	16
Fig. 15	Synchronize recording control.....	19
Fig. 16	Menu item Manage synchronization configuration .....	20
Fig. 17	Configure synchronization configurations .....	20
Fig. 18	Create synchronization configuration.....	21
Fig. 19	Selection of optional software components.....	22
Fig. 20	Single-server system.....	23
Fig. 21	Multi-server system with 2 servers and all-in-one recording .....	24
Fig. 22	Multi-server system with 3 servers, multi-server recording and separate Enterprise Core with database .....	24
Fig. 23	Multi-server system with all-in-one recording, separate Enterprise Core and separate database .....	24
Fig. 24	Multi-server system with All-in-one Parallel Recording and separate failover database .....	25
Fig. 25	Multi-server system with 4 servers with All-in-one Failover Recording, redundant Enterprise Core and external redundant database .....	26
Fig. 26	Multi-server system with 5 servers, multi-server parallel recording and separate Enterprise Core with database .....	26
Fig. 27	Principle single-server system with single core.....	27
Fig. 28	Principle Multi-server system with multi-cores .....	28
Fig. 29	Principle Multi-server system with multi-cores with all redundant components .....	29
Fig. 30	Redundancy solution for PostgreSQL databases .....	30
Fig. 31	Failover operation for 2 Enterprise Cores and redundant database .....	30
Fig. 32	Recording architecture with redundant recording modules.....	31
Fig. 33	Recording architecture with redundant recording modules and recording control modules.....	31
Fig. 34	Single-server system.....	32
Fig. 35	System with 2 servers and 1 all-in-one recording architecture .....	33
Fig. 36	System with 2 servers and All-in-one Failover .....	34

---

Fig. 37	Multi-server system with 4 servers, All-in-one Failover Recording, redundant Enterprise Core and external redundant database.....	35
Fig. 38	System with 2 servers and All-in-one Parallel Recording .....	37
Fig. 39	System with 3 servers and All-in-one parallel Recording.....	38
Fig. 40	System with 4 servers and All-in-one Parallel Recording .....	39

List of tables

Tab. 1    Conversation types ..... 43



---

## 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 either consist of a screen video or of a camera video.

---

**VoIP**

Voice over IP