Nutzer der einheitlichen Rechnerzentralensoftware

Anwenderforderungen Datenverteiler

Version	7.0
Stand	30.09.2016
Produktzustand	akzeptiert
Datei	AFo_DaV_FREI_V7.0_D2016-09-30.doc

Projektkoordinator	NERZ e.V.
Projektleiter	NERZ e.V.
Projektträger	NERZ e.V www.nerz-ev.de
Ansprechpartner	FTB des NERZ e.V.

0 Allgemeines

0.1 Lizenzen

0.1.1 Ursprüngliche Dokumente

Dieses Dokument basiert auf den Inhalten folgender Dokumente (und ggf. Vorgängerversionen):

- AFo_BSVRZ-Gesamt_FREI_V6.0_D2009-11-03.doc ([AFoBSVRZGesamt]).

0.2 Verteiler

Organisationseinheit	Name	Anzahl Kopien	Vermerk
NERZ e.V.		1	

Tabelle 0-1: Dokumentenverteiler

0.3 Änderungsübersicht

Version	Datum	Kapitel	Bemerkungen	Bearbeiter
6.1	15.03.2015	alle	Umstellung der Gesamt-AFo auf Einzel-AFo	H. C. Kniß (HCK), FTB NERZ
6.2	26.01.2016	alle	Anpassungen entsprechend den Ergebnissen des Zertifizierungszwischen-berichts	H. C. Kniß (HCK), FTB NERZ
6.3	27.09.2016	alle	Schlussredaktion	H. C. Kniß (HCK), FTB NERZ
7.0	30.09.2016		Überführung in den Zustand akzeptiert	H. C. Kniß (HCK), FTB NERZ

Tabelle 0-2: Änderungsübersicht

0.4 Inhaltsverzeichnis

0.5 Abkürzungsverzeichnis

Siehe [AbkBSVRZ].

Darüber hinaus werden folgende Abkürzungen verwendet:

BASt	Bundesanstalt für Straßenwesen
BSVRZ	Basis System VRZ

ERZ	Einheitliche Rechnerzentralensoftware
NERZ	Nutzer der ERZ, siehe auch www.nerz-ev.de

0.6 Definitionen

Siehe [GlossarBSVRZ].

Darüber hinaus werden folgende Definitionen verwendet:

--- ---

0.7 Referenzierte Dokumente / URLs

Die folgende Tabelle listet die im Dokument verwendeten Referenzen auf. Zum aktuellen Zeitpunkt sind die folgenden Archiv-URLs vorhanden:

- NERZ-Archiv: http://www.nerz-ev.de/ → Dokumente und Software

[VMOD97]	Der Bundesminister des Inneren, Entwicklungsstandard für IT-Systeme des Bundes Vorgehensmodell, Juni 1997, KBSt, Koordinations- und Beratungsstelle der Bundesregierung für Informationstechnik in der Bundesverwaltung.
[AbkBSVRZ]	Abkürzungsverzeichnis BSVRZ Gesamt NERZ-Archiv: Abk_BSVRZ-Gesamt_FREI_V4.0_D2006-08-15.doc
[GlossarBSVRZ]	Glossar BSVRZ Gesamt NERZ-Archiv: SE-02.0002-Glos-0.4__Glossar__global__.pdf
[AFoBSVRZGesamt]	Anwenderforderungen des BSVRZ, ursprüngliche Gesamtfassung NERZ-Archiv: AFo_BSVRZ-Gesamt_FREI_V6.0_D2009-11-03.doc
[SysArcERZ]	Systemarchitektur zur ERZ NERZ-Archiv: SysArc_ERZ-Gesamt_FREI_V7.0_D2016-09-30.doc
[AFoERZGlobal]	Übergeordnete Anwenderforderungen zur ERZ NERZ-Archiv: AFo_ERZ-Global_FREI_V7.0_D2016-09-30.doc
[AFo_VeW-Sim]	Anwenderforderungen zur ERZ zur "Simulation" NERZ-Archiv: AFo_VeW-Sim_FREI_V7.0_D2016-09-30.doc

0.8 Abbildungsverzeichnis

0.9 Tabellenverzeichnis

1 Zweck des Dokuments

Das vorliegende Dokument beschreibt die Anforderungen an die interne Kommunikationsschnittstelle des Systems zwischen den einzelnen (Fach-) Applikationen.

2 Ist-Aufnahme und Ist-Analyse

Siehe [AFoERZGlobal]

3 IT-Sicherheitsziel

Die Anwenderforderung **GLO-1** (Systemverfügbarkeit), **IT-S-1** (Kommunikation getrennter Systemkomponenten) und **IT-S-2** (Zugriff auf Systemteile) aus dem Dokument [AFoERZGlobal] gilt für die in diesem Dokument beschriebenen Funktionen und ist entsprechend umzusetzen.

GLO-1, IT-S-1 und IT-S-2 *IT-Sicherheitsziel*

4 Bedrohungs- und Risikoanalyse

Die Anforderungen **IT-S-3 bis IT-S-8** aus dem Dokument [AFoERZGlobal] gelten auch für die in diesem Dokument beschriebenen Funktionen.

IT-S-3 bis IT-S-8 *Bedrohung und Risiko*

5 IT-Sicherheit

Die Anwenderforderung ***GLO-2*** *(Zugriffschutz)* und **IT-S-**7 (Unberechtigter Zugriff auf Komponenten) aus dem Dokument [AFoERZGlobal] gilt für die in diesem Dokument beschriebenen Funktionen und ist entsprechend umzusetzen

GLO-2 und IT-S-7 *IT-Sicherheit*

6 Fachliche Anforderungen

6.1 Grobe Systembeschreibung

Der Datenaustausch der einzelnen Systemkomponenten soll über eine einheitliche Schnittstelle realisiert werden. Dadurch soll eine bessere Skalierbarkeit des Systems durch eine konfigurierbare Verteilung der Prozesse möglich werden. Die zeitnahe Weiterverarbeitung von Daten muss durch die ereignisgesteuerte Schnittstelle ermöglicht werden. Um eine weitest gehende Unabhängigkeit der Systemprozesse zu erreichen muss die Schnittstelle so gestaltet sein, dass ein Prozess nicht wissen muss wer seine Eingabedaten liefert und wer seine Ausgabedaten weiterverarbeitet.

Durch den Funktionsblock Datenverteiler wird die systeminterne Kommunikationsschnittstelle (auch über mehrere Rechner, UZ-VRZ, UZ-Bedienstation etc.) realisiert. Der Datenverteiler sorgt dabei für den Weiterleitung und Verteilung aller im System zwischen den Funktionseinheiten auszutauschenden Daten. Eine Funktionseinheit, z.B. die Funktion der „Externen Kommunikation“, stellt die von extern empfangen Daten nach entsprechender Konvertierung dem Datenverteiler zur Verfügung. Dieser reicht die Daten dann an weitere Funktionseinheiten weiter, die ihre Ergebnisse wiederum über den Datenverteiler an andere Funktionseinheiten weiterleitet.

Die spezifischen Anwenderforderungen sind in Kap. 6.5 und Kap 6.6 enthalten.

6.2 Organisatorische Einbettung

Siehe [AFoERZGlobal]

6.3 Nutzung

Siehe [AFoERZGlobal]

6.4 Kritikalität des Systems

Entsprechend den Festlegungen zur Kritikalität in [AFoERZGlobal] wird die Kritikalität für die in diesem Dokument beschriebene Funktionalität als

- mittel

eingestuft

6.5 Externe Schnittstellen

6.5.1 Mensch-Maschine-Schnittstelle

Das spezifizierte Teilsystem ist ein Serversystem bzw. Serverprozess und hat keine Bedienoberfläche.

6.5.2 Externe Kommunikation

6.5.2.1 Externe Standardschnittstelle/Weitere Schnittstellen

DaV-2 *Standardisierte externe Schnittstelle*

Sonstige Kommunikationspartner müssen, soweit dies möglich ist, über das Datenverteiler-Protokoll (siehe Kapitel 6.6.1.1 „Datenverteiler **-** Applikationsfunktionen“) an das System angebunden werden. Dies bedeutet, dass als **standardisierte externe Schnittstelle** die Kommunikationsabläufe und Telegramme des Datenverteilers definiert werden. Damit ist auch der maximale Daten- und Funktionsumfang vollständig spezifiziert.

Einem extern anzubindenden Partner werden dazu die „Applikationsfunktionen“ in Form einer Softwarebibliothek zu Verfügung gestellt. Mittels dieser Funktionen erfolgt auf dem externen Gerät eine Umsetzung in das dort verwendete Protokoll. Die zur Verfügung gestellten Applikationsfunktionen gewährleisten, dass auch bei fehlerhafter Protokollumsetzung auf dem externen Gerät keine Auswirkungen innerhalb des Systems auftreten.

ARC-1 *Weitere externe Schnittstellen*

Ist eine Anbindung von externen Partnern nach diesem Schema nicht durchführbar (dass also die Umsetzung auf dem anzubindenden Gerät ohne Eingriff in die Software der VRZ erfolgt), weil z.B. ein Eingriff in das externe System nicht mehr möglich ist, muss auf Seiten des ERZ-Systems eine **neue** SWE erstellt werden, die eine Umsetzung des externen Protokolls in das systeminterne Protokoll (Datenverteiler) vornimmt.

Die geforderten Funktionen haben keine externen Schnittstellen im Sinne der [SysArcERZ][1].

6.6 Beschreibung der Funktionalität

6.6.1 Datenverteiler

DaV-4 *Allgemeine Funktions-anforderung*

Der Datenaustausch der einzelnen Systemkomponenten soll über eine einheitliche Schnittstelle realisiert werden. Dadurch soll eine bessere Skalierbarkeit des Systems durch eine konfigurierbare Verteilung der Prozesse möglich werden. Die zeitnahe Weiterverarbeitung von Daten muss durch die ereignisgesteuerte Schnittstelle ermöglicht werden. Um eine weitestgehende Unabhängigkeit der Systemprozesse zu erreichen muss die Schnittstelle so gestaltet sein, dass ein Prozess nicht wissen muss wer seine Eingabedaten liefert und wer seine Ausgabedaten weiterverarbeitet.

[1] Die Aufteilung der ursprünglichen Anwenderforderungen [AFoBSVRZGesamt] in einzelne (Teil-) Anwenderforderungen erfolgte im Rahmen der Zertifizierung der ERZ-Software, um eine bessere Pflege und Wartung der Software und der entsprechenden Dokumente zu gewährleisten. Zu diesem Zeitpunkt war die Systemarchitektur bereits vorhanden, die Aufteilung der ursprünglichen AFo-Gesamtfassung erfolgte entsprechend der Systemarchitektur. Aus diesem Grund wird hier auf die [SysArcERZ] verwiesen, da sowohl im Rahmen der AFo-Aufteilung als auch für zukünftige Erweiterungen des Systems die dort festgelegte Systemarchitektur vorgeben ist.

Durch den Funktionsblock Datenverteiler wird die systeminterne Kommunikationsschnittstelle (auch über mehrere Rechner, UZ-VRZ, UZ-Bedienstation etc.) realisiert. Der Datenverteiler sorgt dabei für den Weiterleitung und Verteilung aller im System zwischen den Funktionseinheiten auszutauschenden Daten. Eine Funktionseinheit, z.B. die Funktion der „Externen Kommunikation“, stellt die von extern empfangen Daten nach entsprechender Konvertierung dem Datenverteiler zur Verfügung. Dieser reicht die Daten dann z.B. an die „Datenübernahme und -aufbereitung“ weiter, die ihre Ergebnisse wiederum über den Datenverteiler an andere Funktionseinheiten weiterleitet. Eine detaillierte Beschreibung der Anforderungen an den Datenverteiler enthalten die nachfolgenden Kapitel.

DaV-5 *Client-Server Architektur des Datenverteilers*

Der **Datenverteiler** stellt das Bindeglied zwischen allen Applikationen sowie zwischen den Applikationen und der Konfiguration dar.

Die einzelnen Applikationen stehen mit dem Datenverteiler in einem Client **-** Server Verhältnis, wobei die Applikationen die Clients und der Datenverteiler den Server darstellt.

DaV-6 *Transparenter Zugriff bei Verteilung auf mehrere Rechner*

Wie bereits beschrieben ist selbst bei Verteilung des Systems auf mehrere Rechner der Zugriff für die einzelnen Applikationen auf den Datenverteiler transparent, d.h. dass sie sich nicht darum kümmern müssen, wie die Daten zwischen den einzelnen Rechnern transportiert werden. Diese Aufgabe übernimmt der Datenverteiler.

Innerhalb dieses Dokuments wird immer von **dem** (einem) Datenverteiler geredet. Dies ist Sicht der Applikationen, da sie nur einen Datenverteiler **-** dessen lokale Instanz sich auf demselben Rechner wie die Applikation befindet **-** kennen und das Zusammenspiel der Gesamtheit aller Datenverteiler für sie transparent ist. Dies bedeutet, dass in diesem Dokument immer die Gesamtheit aller Datenverteiler des Systems (je Rechner ein Datenverteiler) bei Betrachtungen aus Sicht der Applikationen (und das ist die Regel) als **der** Datenverteiler bezeichnet wird. Lediglich bei Betrachtungen zur Kommunikation zwischen den lokalen Datenverteilern werden diese explizit unterschieden. Dieses Zusammenspiel der lokalen Datenverteiler beschreibt das Kapitel 6.6.1.2 „Datenverteiler **-** Server“. Das Kapitel 6.6.1.1 „Datenverteiler **-** Applikationsfunktionen“ beschreibt die applikationsseitigen Funktionen des Segments Datenverteiler. Die folgende Abbildung skizziert noch einmal den Zusammenhang.

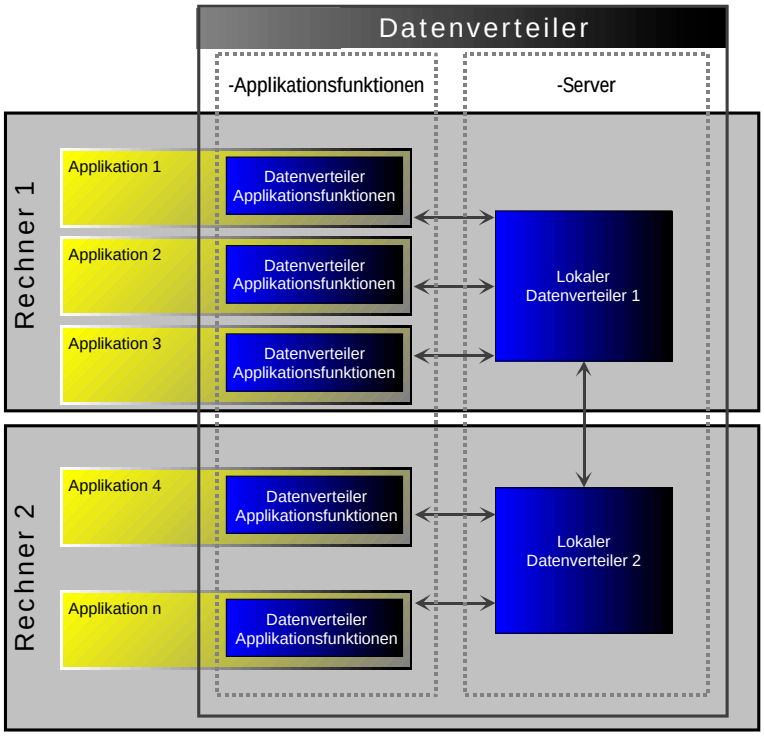

Abbildung 6-1: Datenverteiler (lokale und globale Sicht)

6.6.1.1 Datenverteiler - Applikationsfunktionen

DaV-7 *Clientfunktionen des Datenverteilers*

Die Clients des Datenverteilers sind die Applikationen. Da alle Applikationen ein identisches systemtechnisches Verhalten bezüglich des Datenverteilers (Server) besitzen **müssen**, ist der Clientanteil des Segments Datenverteiler in Form einer Softwarebibliothek zur Verfügung zu stellen. Dieser Clientanteil des Datenverteilers stellt innerhalb der Applikation das Bindeglied zwischen dem Datenverteiler (Server) und der eigentlichen Applikation dar. Bei der Applikationsentwicklung ist diese Library einzubinden und die dadurch zur Verfügung gestellte Schnittstelle zum Datenverteiler zu verwenden. Im Folgenden wird immer von **Datenverteiler - Applikationsfunktionen** gesprochen, wenn diese Funktionen gemeint sind.

DaV-8 *abzudeckender Funktionsumfang*

Dabei werden folgende grundsätzliche Funktionen zur Verfügung gestellt:

- Aufbau einer Verbindung zum lokalen Datenverteiler
- Versenden von Telegrammen
- Abfrage von anliegenden Telegrammen (Empfang)
- Automatische Quittierung von empfangenen Telegrammen
- Automatische Wiederholung von Sendeversuchen bei Übertragungsfehlern
- Automatische Abwicklung des Handshake zwischen Client und Server
- Abbau einer Verbindung zum Datenverteiler
- Automatisches Versenden von Keep-Alive Telegrammen an den Server (Datenverteiler)
- Priorisierung von Telegrammen

DaV-9 *Unterscheidung zwischen System- und Datentelegrammen*

Mittels dieser Funktionen werden dann die Daten über genormte Telegramme ausgetauscht, wobei zwischen **Systemtelegrammen** und **Datentelegrammen** unterschieden wird. Dabei gilt folgende Struktur:

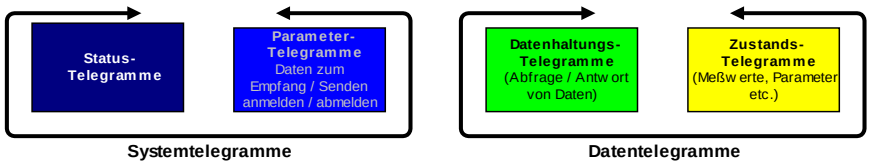

Abbildung 6-2: Klassifizierung der Telegrammtypen

Systemtelegramme dienen der direkten Kommunikation mit dem Datenverteiler (Server) und werden direkt vom Datenverteiler verarbeitet bzw. beantwortet. Zu den Systemtelegrammen gehören:

DaV-10 *Parametertelegramme*

- **Parameter**telegramme. Mit den Parametertelegrammen parametriert die Applikation den Datenverteiler entsprechend ihren Anforderungen. D.h.,
  - sie **meldet Daten an**. Damit teilt eine Applikation dem Datenverteiler mit, welche Datentelegramme sie für welche Objekte empfangen will (Eingabedaten der Applikation) und welche Datentelegramme sie für welche Objekte zur Verfügung stellt (Ausgabedaten der Applikation). Die Filterung bzgl. des Datenverkehrs durch den Datenverteiler ist also je Telegrammtyp (Datenart, Attribut) je Objekt möglich. Damit kann eine Applikation z.B. gezielt die Daten „Temperatur in 30cm Tiefe“ für ein einziges DE anfordern oder nur die „Verkehrswerte der Version 3“ für eine spezielle Schleife liefern.
  - sie **meldet Daten ab**. Damit werden die mittels der Parametertelegramme beim Datenverteiler angeforderten / zu liefernden Daten wieder abgemeldet. Die Abmeldung ist ebenfalls gezielt je Telegrammtyp je Objekt möglich. An- und Abmeldung von Telegrammen (applikationsabhängige Parametrierungen des Datenverteilers) erfolgen dynamisch, so dass

eine Applikation abhängig von ihrem momentanen Datenbedarf / Datenlieferumfang den Datenverteiler entsprechend umparametrieren kann.

DaV-11 *Status-telegramme*

- **Status**telegramme an den Datenverteiler. Damit können Statusinformationen vom Datenverteiler (von allen im System vorhanden Datenverteilern) abgefragt werden. Mögliche Statusabfragen sind:
  - Abfrage der lokal / im System angemeldeten Applikationen und die von diesen Applikationen angeforderten / lieferbaren Daten. Diese Informationen sind nur für spezielle Überwachungsapplikationen von Interesse. Normale Applikationen benötigen diese Informationen nicht und sollten sie auch nicht verwenden!

DaV-12 *Standardisierter Ablauf Applikation-Datenverteiler*

Die Schnittstelle zwischen Applikationen und den Datenverteiler-Applikationsfunktionen muss den im Folgenden beschriebenen Aufbau einer normalen Applikation ermöglichen und unterstützen. (Darstellung hier zur Vereinfachung ohne Fehler- und Ausnahmebehandlung!):

- **Start der Applikation**
  - Aufbau einer Verbindung zum lokalen Datenverteiler (durch die integrierte Datenverteiler-Applikationsfunktionen des Datenverteilers)
- **Initialisierung**
  - Abfrage der relevanten Konfigurationsdaten für die Applikation (Daten - Datenhaltungstelegramme)
  - Interne Initialisierung der benötigten Datenstrukturen, Zusammenstellung der benötigten Daten bzw. der durch die Applikation gelieferten Daten.
  - Parametrierung des Datenverteilers durch die Applikation
    - Anmeldung der durch die Applikation benötigten Daten beim Datenverteiler (System - Parametertelegramme)
    - Anmeldung der durch die Applikation gelieferten Daten beim Datenverteiler (System - Parametertelegramme)
    - Warten, bis alle angeforderten Daten als verfügbar gemeldet worden sind, dann
- **Hauptroutine als (Endlos)Schleife**
  - Abfrage (Empfang) von anliegenden Telegrammen (Daten - Zustandstelegrammen). Die Abfrage erfolgt über die entsprechenden Funktionen der Datenverteiler - Applikationsfunktionen innerhalb der Applikation. Der Datenverteiler verschickt angemeldete Daten spontan an die Applikationen.
  - Verarbeitung der Eingangsdaten mit dem **applikationsspezifischen Algorithmus**.
  - Versenden der neuen Ergebnisdaten an den Datenverteiler (Daten - Zustandstelegrammen) über die entsprechenden Funktionen der Datenverteiler - Applikationsfunktionen.
  - (optional Umparametrierung des Datenverteilers bei Bedarf).
- **Ende der Applikation (optional, nur bei Applikationen mit definiertem Ende)**
  - (Abmeldung aller Telegramme beim Datenverteiler mit System - Parametertelegrammen)
  - Abmeldung der Applikation (dadurch werden intern im Datenverteiler alle noch offenen Anmeldungen der Applikation ordnungsgemäß abgemeldet und die Applikation aus der internen Verwaltung des Datenverteilers gelöscht).

6.6.1.1.1 Verbindungsaufbau zum Datenverteiler

DaV-13 *Verbindungsaufbau zum Datenverteiler*

Eine Applikation baut über die entsprechende Funktion der Applikationsfunktionen des Datenverteilers eine Verbindung zum Datenverteiler (Server) auf.

Gelingt der Verbindungsaufbau nicht, dann liefert die Funktion eine entsprechende Fehlermeldung zurück, so dass die Applikation geeignet reagieren kann.

Nach dem Verbindungsaufbau muss die Applikation sich als erstes gegenüber dem Datenverteiler (Server) authentifizieren. Über diese Authentifizierung werden die Zugriffsrechte der Applikationen festgelegt.

6.6.1.1.2 Parametrierung des Datenverteilers

DaV-14 *Parametrierung des Datenverteilers*

Jede Applikation muss, nachdem sie eine Verbindung zum lokalen Datenverteiler aufgebaut hat, „ihre“ Daten anmelden. Dies bedeutet, dass sie sowohl

- alle Daten, die sie vom Datenverteiler erhalten will (Eingangsdaten der Applikation), als auch
- alle Daten, die sie potentiell an den Datenverteiler sendet[2] (Ausgangsdaten der Applikation)

vollständig mitteilt.

Die Applikation **parametriert** damit ihren Datenverteiler.

Diese Parametrierung geschieht über **System - Parametertelegramme**, da diese ja vom Datenverteiler direkt verarbeitet werden. Folgende Angaben müssen gemacht werden:

- **ObjektID**

  Durch die ObjektID wird festgelegt, für welches Objekt (z.B. DE 7 an der Streckenstation 5 der Unterzentrale Irgendwo, die Anlage VBA A3 Fahrtrichtung Süd, eine spezielle USV der VRZ etc.) Daten geliefert werden sollen. Die angegebenen Objekte müssen hier explizit referenziert werden. Objekte aus hierarchische Zusammenhängen (z.B. alle MQs der VBA A3 Fahrtrichtung Süd) müssen aus der Konfiguration bestimmt werden und sind dem Datenverteiler nicht bekannt.

- **TelegrammID**

  Die TelegrammID spezifiziert einen speziellen Telegrammtyp und damit eine festgelegte Kombination von Attributen. Zusammen mit der ObjektID wird dadurch eindeutig ein Objekt mit seinen (einem Teil seiner) Attributen beschrieben.

- **Richtung**

  Durch das Flag Richtung wird lediglich festgelegt, ob die Applikation die Daten an den Datenverteiler sendet (**SENDEN**) oder von diesem empfangen will (**EMPFANGEN**).

- **Quittung**

  Über das Flag Quittung kann erreicht werden, dass zu den versandten oder empfangenen Daten jeweils durch das Archivsystem ein Quittierungstelegramm erzeugt wird, sobald die Daten gespeichert wurden. Dies kann u.a. dazu benutzt werden, um empfangene Daten erst dann weiterzuverarbeiten, wenn die Verarbeitung durch die Archivierung durch die Datenhaltung quittiert wurde. Die Quittierung wird später ausführlich gesondert behandelt.

An- und Abmeldungen sind dynamisch zur Laufzeit beliebig durchführbar, eine Applikation kann also ihren Datenverteiler dynamisch umparametrieren. Die Visualisierung wird z.B. immer nur die Daten anfordern, die zur Darstellung der gerade geöffneten Fenster benötigt werden. Wird ein Fenster geschlossen, kann sie die entsprechenden Daten gezielt abmelden, wird ein Fenster geöffnet entsprechend anmelden.

[2] Auch potentiell zu versendende Daten, zu denen kein Zustand ermittelbar ist, werden angemeldet.

6.6.1.1.3 Betrieb des Datenverteilers

Mit der Anmeldung, welche Daten die Applikation zur Verfügung stellt, wird das aktuelle Datum oder für den Fall, dass der aktuelle Zustand noch nicht bekannt ist, eine entsprechende Kennung mitgeliefert. Nach der Anmeldung kann die Quellapplikation neue Datensätze an den Datenverteiler senden. Wenn die Quellapplikation den aktuellen Zustand nicht mehr ermitteln kann sendet sie statt den Daten eine entsprechende Kennung. Daten bzw. Kennung werden vom Datenverteiler an die Applikationen versendet, die sich hierauf angemeldet haben.

6.6.1.1.4 Kommunikationsüberwachung

DaV-15 *Kommunikationsüberwachung*

Die Kommunikation zwischen Applikationen und Datenverteiler ist über ein gesichertes Kommunikationsprotokoll abzuwickeln.

Dabei läuft die Sicherung auf den unteren OSI - Ebenen automatisch ab, d.h. für die Applikation transparent. Die entsprechenden Funktionalitäten sind in den **Datenverteiler - Applikationsfunktionen** gekapselt. Dabei werden folgende Abläufe unterstützt:

Überwachung beim Senden und Empfangen von Daten:

DaV-16 *Telegrammquittung*

- Das eingesetzte Kommunikationsprotokoll muss Fehlererkennungsmechanismen und eine Quittierungsmechanismus enthalten, mit denen Übertragungsfehler weitgehend erkannt und durch erneute Sendeversuche korrigiert werden. Schlägt ein Sendeversuch mehrmals fehl bzw. wird eine maximale Timeoutzeit überschritten, gilt die Verbindung als gestört und ist verwaltungstechnisch vollständig abzubauen. Wird das Telegramm quittiert, kann das nächste Telegramm versendet werden.

- Applikation und Datenverteiler dürfen sich nur dann Daten schicken, wenn sie **empfangsbereit** sind. Die Kommunikationspartner dürfen temporär auf **nicht empfangsbereit** (z.B. bei vollem Empfangspuffer) schalten. Die zu versenden Telegramme sind senderseitig solange zu puffern. Entsteht senderseitig ein Pufferüberlauf gilt die Verbindung als gestört und ist verwaltungstechnisch vollständig abzubauen. Der Puffer ist zu löschen. Ein Überschreiben des Puffers ist nicht zulässig, da dies zu einer unkontrollierbaren Datenlücke auf Empfangsseite führt.

DaV-18 *Datenpufferung*

Aktive Überwachung des Kommunikationspartners:

DaV-17 *aktive Überwachung des Kommunikationspartners*

- Zur aktiven Überwachung des Kommunikationspartners sind folgende Mechanismen vorzusehen:
  - Keep-Alive: Um Verbindungsfehler auch dann zu erkennen, wenn keine Daten übertragen werden, werden von beiden Kommunikationspartnern automatisch Keep-Alive-Telegrammen versendet. Wenn über eine bestimmte Zeit kein Keep-Alive-Telegramm empfangen wurde, ist von einem Kommunikationsproblem auszugehen.
  - Telegrammlaufzeitermittlung: Zur Ermittlung der Telegrammlaufzeit zwischen zwei Kommunikationspartnern wird ein entsprechendes Telegramm versendet, das von der Gegenseite entsprechend quittiert wird. Aus der Zeit die zwischen Versand des Telegramms und Empfang der Quittung vergangen ist wird die Telegrammlaufzeit ermittelt.

6.6.1.1.5 Priorisierung von Telegrammen

DaV-19 *Priorisierung von Telegrammen*

Der Datenverteiler unterstützt die priorisierte Versendung von Telegrammen. Die Telegramme werden dazu entsprechend ihrer Priorität in den Sendepuffer bzw. Empfangspuffer eingetragen. Dabei werden Telegramme mit höherer Priorität an den Anfang gestellt, wobei Telegramme mit gleicher Priorität ihre Sende- bzw. Empfangsreihenfolge beibehalten. Innerhalb einer Prioritätsstufe ist die natürliche Reihenfolge zu garantieren (FIFO-Prinzip), ansonsten werden die Telegramme entsprechend ihre Prioritätenklasse abgearbeitet.

Die Priorisierung erfolgt nicht durch die Applikation, sondern wird über Telegrammklassen[3] als konfigurierendes Datum des Datenverteilers global festgelegt. Die Festlegung **muss systemweit eindeutig sein!**

6.6.1.2 Datenverteiler - Server

DaV-20 *Serverfunktionen des Datenverteilers*

Die Schnittstelle zum Client (den Applikationen) ist durch die Beschreibung der Clientseite festgelegt. Aus Sicht der Clients existiert ein „großer" Server (Datenverteiler), an dem alle Applikationen angeschlossen sind. Damit diese aus Applikationssicht sehr einfache und klare Struktur auch funktioniert, sind auf Serverseite einige Anstrengungen notwendig. Die einzelnen, lokalen Datenverteiler müssen - um sich nach außen wie **ein** Datenverteiler zu verhalten - folgende Aufgaben abwickeln:

DaV-21 *Auf- und Abbau von Verbindungen zwischen Datenverteilern*

- Automatischer Auf- und Abbau der Verbindung zwischen den lokalen Datenverteilern. Dazu kennt jeder lokale Datenverteiler die für ihn relevanten[4] anderen Datenverteiler (Kommunikationspartner). Ein Datenverteiler kann dabei mehreren anderen Datenverteilern zugeordnet werden (mit Priorisierung), womit sich eine erhöhte Ausfallsicherheit des Systems beim Ausfall einzelner Rechnerverbindungen ergibt. Der Versuch, eine Verbindung aufzubauen, erfolgt zyklisch bis zum Erfolg.

---

[3] Online-Daten, nachgelieferte Daten, aus dem Archiv abgefragte Daten und Systemtelegramme sind Beispiele Telegramme unterschiedlicher Telegrammklassen.

[4] Die „relevanten anderen Datenverteiler" werden jeweils in der Datenhaltung vorgehalten. Eine Referenz auf die vom Datenverteiler zu verwendende Datenhaltung wird beim Start übergeben.

- Meldet eine Applikation Daten zum Empfang vom Datenverteiler an, die dieser aber nicht liefern kann, weil bei ihm keine Applikation als Lieferant der relevanten Daten angemeldet ist, so muss der Datenverteiler automatisch die jeweiligen Datenquellen ermitteln. Dabei müssen die Datenquellen vom Datenverteiler selbständig und ohne spezielle Konfiguration ermittelt werden. Wenn keine Quelle für die angemeldeten Daten bestimmt werden kann bleibt die Anmeldung bestehen und die Applikation erhält eine entsprechende Kennung. Sobald die Datenquelle zur Verfügung steht, leitet der Datenverteiler die Daten an die anfordernde Applikation weiter.

DaV-22 *automatische Datenquellen-ermittlung*

- Versendet ein Datenverteiler an eine Applikation Daten, die er selbst bei einem anderen Datenverteiler erhält und fällt die Verbindung zwischen den Datenverteilern aus, so sucht der Datenverteiler selbständig nach alternativen Wegen zum Empfang der Daten. Der Applikation wird die kurzfristige Unterbrechung bzw. der Wiederaufbau der Verbindung mitgeteilt, damit diese darauf geeignet reagieren kann.
- Wenn die Verbindung zu einem Datenverteiler ausgefallen ist, dann wird in regelmäßigen Abständen versucht die Verbindung wiederherzustellen. Nach einem erfolgreichen Wiederaufbau der Verbindung wird von den (durch den Verbindungsausfall bedingten) alternativen Datenvermittlungswegen auf die normalen Wege zurückgeschaltet.
- Die Verbindung zwischen den einzelnen Datenverteilern ist gesichert.

6.6.1.2.1 Parametrierung des Datenverteilers

Die Datenverteiler parametrieren die ihnen zugeordneten Datenverteiler (Konfiguration!) bei Bedarf ähnlich wie dies auch die Applikationen tun. Dabei kann jeder Datenverteiler gleichzeitig sowohl Client als auch Server eines anderen Datenverteilers sein. Die Parametrierung eines Datenverteilers heißt auch hier, dass an diesen Datenverteiler mittels Systemtelegrammen Datenanforderungen gestellt werden.

Diese Parametrierung geschieht über **System - Parametertelegramme**, wie sie bei der Parametrierung der Datenverteiler durch Applikationen beschrieben wurden.

DaV-23 *Parame-trierung über Systemtele-gramme*

6.6.1.2.2 Datenvermittlung

Die eigentliche Datenvermittlung - d.h. der gesamte Datentransport im System - läuft dabei kaskadiert ab. Dies bedeutet das die Daten über mehrere Stufen verteilt werden können. Dabei muss berücksichtigt werden, dass keine Zyklen im Datenverkehr entstehen. Kann kein Datenverteiler die Daten liefern, erhält die Applikation eine negative Quittung.

DaV-24 *Datenvermittlung und Datenverteilung*

Die eigentliche Datenverteilung läuft, sobald die Parametrierungen der Datenverteiler durch die Applikationen nach den zuvor beschriebenen Mechanismen erfolgt ist, wie im Folgenden beschrieben ab.

DaV-25 *Prüfungen bei der Datenver-mittlung*

- Die Datenverteiler erhalten von ihren Applikationen spontan die parametrierten Daten. Die Datenverteiler überprüfen jeweils, ob
  - die empfangen Daten von der Applikation überhaupt ordnungsgemäß parametriert worden sind.
  - ob die Applikation überhaupt berechtigt ist, die Daten zu verschicken (Zugriffsrechte der Applikationen).
- Sind diese Kriterien erfüllt (sonst werden die Daten zurückgewiesen), versenden die Datenverteiler die Daten an die darauf angemeldeten Applikationen und andere darauf angemeldete Datenverteiler, sofern
  - die Zielapplikation berechtigt ist, die Daten zu empfangen (Zugriffsrechte der Applikationen).

Die Prüfung der Zugriffsrechte der Applikationen erfolgt erst zur Laufzeit und nicht schon bei der Parametrierung der Datenverteiler, da die Zugriffsrechte auch zur Laufzeit geändert werden können und damit vom Datenverteiler entsprechend berücksichtigt werden müssen.

6.6.1.2.3 Kommunikationsüberwachung

DaV-26 *Kommunikationsüberwachung zwischen Datenverteilern*

Die Verbindung zwischen den einzelnen Datenverteilern ist gesichert, das heißt:

- Die einander zugeordneten Datenverteiler schicken sich zyklisch „Keep alive“ Telegramme, die von der Gegenseite beantwortet werden (siehe oben). Damit lassen sich durch jeden Datenverteiler unabhängig vom inhaltlichen Datenverkehr seine Kommunikationspartner und die Kommunikationsstrecken überwachen.
- Alle versendeten Telegramme werden von der Gegenstelle quittiert.
- Nicht quittierte Telegramme werden eine parametrierbare Anzahl mal wiederversandt. Danach wird die Gegenseite für „tot“ erklärt, die von diesem Datenverteiler angemeldeten Daten aus den internen Verwaltungsstrukturen gelöscht und die von diesen Änderungen betroffenen Applikationen benachrichtigt.
- Datenverteiler, die ordentlich beendet werden (z.B. bei Neustart oder bei Wartungsarbeiten) melden sich bei den ihnen zugeordneten Datenverteilern ordnungsgemäß ab, so dass diese ihre Verwaltungsstrukturen aufräumen können.

6.6.1.2.4 Systemfunktionen

DaV-27 *Systemfunktionen des Datenverteilers*

Zu den Systemfunktionen des Datenverteilers gehören alle Funktionen die - über Statustelegramme gesteuert - interne Zustandsinformationen des Datenverteilers verfügbar machen. Damit lässt sich z.B. abfragen,

- welche Applikationen lokal / im System angemeldet sind und die von diesen Applikationen angeforderten / lieferbaren Daten.
- der Kommunikationsstatus der verbundenen Applikationen (noch nicht versandte Telegramme, Zeitverhalten, Fehlerstatistiken etc.).

6.6.1.2.5 Priorisierung von Telegrammen

DaV-28 *Priorisierung von Telegrammen zwischen Datenverteilern*

Die Priorisierung von Telegrammen erfolgt analog wie bei den **Datenverteiler - Applikationsfunktionen** beschrieben.

6.6.1.2.6 Anforderungen an das Kommunikationsnetz

Als Netzwerk- und Verbindungsprotokoll ist TCP/IP einzusetzen. Dieses setzt auf Protokollen auf, die abhängig von der jeweiligen Umgebung und dem eingesetzten physikalischen Netz benötigt werden.

Die systeminterne Kommunikation der verschiedenen Systemkomponenten und die Kommunikation mit externen Systemen erfolgt auf Basis von TCP/IP.

DaV-29 *Netzwerkprotokoll*

Es ist davon auszugehen, dass im zu verwendenden Kommunikationsnetz verschiedene Firewall Konzepte zum Einsatz kommen. Beim Einsatz von Packet-Filtern und Application-Gateways ist zu berücksichtigen, dass eine (möglichst geringe) Menge von fest definierten TCP-Portnummern von den Systemkomponenten für die Kommunikation benutzt wird. Beim Einsatz von Application-Gateways ist weiterhin zu berücksichtigen, dass eventuell spezielle Proxys bereitgestellt werden und in den Kommunikationsprotokollen berücksichtigt werden müssen.

Der Anschluss von Unterzentralen, Bedienstationen und externen Systemen an die VRZ geschieht über Verbindungen mit einem Durchsatz von mindestens 64000 bps.

In einzelnen Fällen (z.B. APW) muss auch ein Anschluss bei lediglich 9.600 bps möglich sein, wobei in diesen Fällen die Bedienbarkeit eingeschränkt seien kann (langsamer, Einschränkung auf lokale Anlage und weniger komplexe Darstellung).

6.6.1.2.7 Sicherheitsanforderungen an Datenverteilerverbindungen

6.6.1.2.7.1 Authentifizierung

DaV-30 *Authentifizierung*

Zur Authentifizierung der Kommunikationspartner (Applikation mit Datenverteilerapplikationsfunktionen an Datenverteiler und Datenverteiler-Datenverteilerkopplung) sind zur Authentifizierung bei der Anmeldung zum Aufbau der Verbindung Benutzerkennung und Passwort zu verwenden.

Bei den zur Authentifizierung eingesetzten Verfahren muss der Zugriff auf die Authentifizierungsmerkmale (das Passwort) nachgewiesen werden, ohne diese selber im Klartext zu übertragen.

Das Passwort muss so verschlüsselt gespeichert werden, dass sich damit die Gültigkeit der Benutzer/Klartextpasswort Kombination ermitteln lässt.

Aus dem verschlüsselten Passwort darf sich das Klartextpasswort nicht ermitteln lassen. Zudem darf das verschlüsselte Passwort nicht zur zum Verbindungsaufbau nutzen lassen.

Bei der Authentifizierung ist doppelt auszuführen, d.h., zuerst prüft der Server die Anmeldedaten des Client und anschließend überprüft der Client die Anmeldedaten des Servers. Nur wenn beide Authentifizierungsrunden erfolgreich sind, wird die Verbindung hergestellt.

6.6.1.2.7.2 Begrenzung der Login-Versuche

DaV-31 *Begrenzung Login-Versuche*

Um Wörterbuch- und Brute-Force-Angriffe über die Datenverteilerschnittstelle, bei denen sehr schnell viele Passwörter durchprobiert werden können, aktiv zu verhindern, muss im Datenverteiler bei fehlgeschlagenen Login-Versuchen eine entsprechende Drosselung realisiert werden. Nachfolgende Login-Versuche sind z. B. um z Sekunden zu verzögert, bis wieder ein erfolgreicher Login-Versuch durchgeführt wurde. Der Wert z kann dabei mit jedem weiteren fehlgeschlagenen Versuch von anfangs 1 Sekunde bis z.B. maximal 60 Sekunden vergrößert werden.

6.6.1.2.7.3 Sicherung der Datenintegrität

DaV-32 *Datenintegrität*

Die Sicherung der Datenintegrität muss es dem Empfänger einer Nachricht ermöglichen zu prüfen, ob die Nachricht tatsächlich vom Absender versendet wurde und nicht auf dem Übertragungsweg modifiziert wurde. Dies ist mit digitalen Signaturen oder mit dem Message Authentication Code-Verfahren (kurz MAC-Verfahren) zu realisieren.

6.6.1.2.7.4 Sicherung der Vertraulichkeit durch Verschlüsselung

DaV-33 *Vertraulichkeit durch Verschlüsselung*

Zur Sicherung der Vertraulichkeit sind Verschlüsselungsalgorithmen zu verwenden. Die zu übertragenden Daten sind mit Hilfe eines Schlüssels in eine Form zu bringen, in der der ursprüngliche Inhalt nicht mehr erkennbar ist. Nur mit dem passenden Schlüssel darf der ursprüngliche Inhalt wieder hergestellt werden können.

6.6.1.2.8 Verwaltung von Simulationsvarianten

DaV-3 *Verwaltung der Simulationsvariante*

Um die dabei mehrfach entstehenden Instanzen der SW-Einheiten und Datenströme unterscheiden zu können, wird im Folgenden eine Simulationsvariante (oder nur Variante) eingeführt, über die die Zusammengehörigkeit von Berechnungsstrecken festgelegt wird.

Dazu übergeben die Applikationen bei der Anmeldung beim Datenverteiler (Verbindungsaufbau) diesem die beim Applikationsstart per Aufrufparameter übergebene Variantennummer für die Simulation (Aufrufparameter `-simVariante`). Über diese Variantennummern werden die Telegramme zusätzlich unterschieden. Das Hinzufügen der Variantennummern zu den Telegrammen geschieht (automatisch) über die Datenverteiler-Applikationsfunktionen bei der Übergabe von Telegrammen an den Datenverteiler. Beim Weiterleiten von Telegrammen an die Applikationen wertet der Datenverteiler die Variante des Telegramms und der anfordernden Applikation aus und überträgt das Telegramm bei Übereinstimmung an diese.

Details zur grundsätzlichen Funktionsweise der Simulation, die in Teilen durch den Datenverteiler funktional bereitgestellt werden muss, enthält [AFo_VeW-Sim].

6.7 Qualitätsforderungen

ARC-4 bis ARC-9, GLO-18 bis GLO-22 und GLO-28
QS-Anforderungen

Es gelten die Qualitätsanforderungen gemäß [AFoERZGlobal] ARC-4 bis ARC-9, GLO-18 bis GLO-22 und GLO-28.

Darüber hinaus gehende Anforderungen an die Qualität werden nicht gestellt.

7 Randbedingungen

7.1 Technische Randbedingungen

7.1.1 Programmiersprache

Als Standardprogrammiersprache ist Java 8 oder höher[5] zu verwenden.

7.2 Organisatorische Randbedingungen

Siehe [AFoERZGlobal]

7.3 Sonstige Randbedingungen

Siehe [AFoERZGlobal]

[5] Es ist in Abstimmung mit dem AG jeweils die im NERZ Systemen aktuell eingesetzte Version zu verwenden. Dies kann eine höhere Version als die aktuell eingesetzte Version Java 8. sein, es muss aber nicht immer die letzte aktuell verfügbare Version sein.

8 Anforderungsverzeichnis