



SPILLEMÝNDIGHEDEN

Date: 24-05-2012

Version: 1.1

Instructions for technical requirements

Standard Records SAFE Tamper Token ROFUS

Gambling Control System
2012

Document history:

Version	Date	Description of changes
0.9	2012.02.23	Draft for hearing.
1.0	2012.03.12	Published official version.
1.1	2012.05.24	<p>The information below was also provided to all Licence Holders by e-mail on May 3, 2012.</p> <p>Details regarding cutover between v1 and v2 Section 1.2.1 inserted with detailed descriptions of the transition between v1 and v2 of standard records.</p> <p>Update of xml-structure FastOddsSlutStruktur In the version of the XSD schemas from March 12, the xml-structure FastOddsSlutStruktur contained the attributten SpilFilErstatningIdentifikation, which is used for replacement data. This was misleading since changes to FastOddsSlutStruktur must be handled as bet-resettlement, as described in section 2.3 Error handling. Therefore, an updated version of the XSD schemas (“Standard Record XSD schemas – 24may2012.zip”) has been provided as can be seen in section 2.2. The only change is the removal of the optional field SpilFilErstatningIdentifikation in FastOddsSlutStruktur.</p>

Table of contents

1. Introduction.....	5
1.1. Version 1 of Standard Records and documentation	5
1.2. Version 2 of Standard Records and documentation	5
2. Standard Records	8
2.1. Conceptual model.....	8
2.2. XSD schemas for Standard Records.....	8
2.2.1. Schema for fixed odds games.....	9
2.2.2. Schemas for casino games.....	9
2.2.3. Schemas for poker cash games.....	10
2.2.4. Schemas for poker tournaments.....	10
2.2.5. Schemas for pool games.....	11
2.2.6. Schemas for manager games	12
2.2.7. Schema for the triggering of a jackpot.....	12
2.2.8. Schema for End of Day reporting.....	12
2.3. Error handling	20
2.3.1. Cancellations	22
2.3.2. Cancellations and the End of Day report.....	22
2.3.3. Bet-resettlement.....	22
2.3.4. Bet-resettlement and the End of Day report	23
2.3.5. Replacement data.....	23
2.3.6. Replacement data and the End of Day report	23
2.3.7. Other error types.....	24
2.3.8. Other error types and End of Day report.....	24
2.4. Release process for Standard Records.....	24
2.4.1. Major releases.....	24
2.4.2. Minor releases	25
3. SAFE	26
3.1. Folderstructure on SAFE	26
3.2. SAFE configurations	27
3.2.1. Naming standards	27
3.2.2. Packing of files.....	27
3.2.3. Choice of date-folder for zip-file.....	28
3.2.4. UTC time for SAFE.....	29
4. Tamper Token and ROFUS	30
4.1. Tamper Token: Webservice description, WSDL- and XSD-files	30
4.2. ROFUS: Webservice descriptions, WSDL- and XSD-files.....	30
4.3. General technical configurations.....	30
4.3.1. Header information and error handling.....	31
4.3.2. Service call – creation of http(s) connection and receiving data	32
4.4. Tamper Token configurations	33
4.4.1. Error handling for Tamper Token Hent service.....	33
4.4.2. Error handling for Tamper Token Luk service.....	34
4.4.3. Mechanism for MAC generation	35
4.4.4. MAC API	35
4.4.5. MAC algorithm – figure.....	37
4.4.6. Example of MAC calculation.....	38
4.4.7. Handling unused tokens	38
4.5. Examples of Tamper Token and ROFUS service request and reponse	39
4.5.1. TamperTokenHent	39

4.5.2.	TamperTokenLuk	40
4.5.3.	GamblerCheck for person nummer not registered in ROFUS.....	41
4.5.4.	GamblerCheck for person nummer temporarily registered in ROFUS	41
4.5.5.	GamblerCheck for person nummer indefinitely registered in ROFUS.....	42
5.	Licence Holder's quality assurance.....	43
5.1.	Checklist for new version of Standard Records	43
5.2.	Schema validation of Standard Records using XML Spy.....	43
5.3.	Validation of correct Tamper Token MAC calculation.....	44

1. Introduction

The purpose of this document is to provide Licence Holder with an overview of the detailed technical requirements for providing online games in Denmark. The document contains requirements which previously have been presented in several different documents on the Danish Gambling Authority's (DGA) website. This document also holds referencens to documents and files which describes the requirements in more details.

The document describes the detailed technical requirements with respect to Standard Records, SAFE, Tamper Token and ROFUS. The document also proposes suggestions to how Licence Holder may perform quality assurance.

1.1. Version 1 of Standard Records and documentation

Version 1 of Standard Records is from October 2011.

1.2. Version 2 of Standard Records and documentation

The Danish Gambling Authority will during February and March 2012 publish version 2 of Standard Records. The new version has the purpose of making Standard Records more robust to data errors and to minimize misunderstandings and misinterpretations.

The new version is a major release as described in section 2.4. Licence Holder has the possibility of providing comments and suggestions to the new Standard Records through a hearing process. The Danish Gambling Authority encourage all Licence Holders to provide input during the hearing process, but cannot guarantee that the input will be incorporated.

There will not be a test periode for transition to version 2 of standard records. Instead, Licence Holder must complete and send the sheet "Ændringer til standard records – 12mar2012.xlsx" to the DGA. The sheet is a check list with a description of all changes and Licence Holder should indicate whether or not the changes have been implemented. Another purpose of creating the check list is to ease the work for Licence Holder in terms of transition to version 2 of standard records.

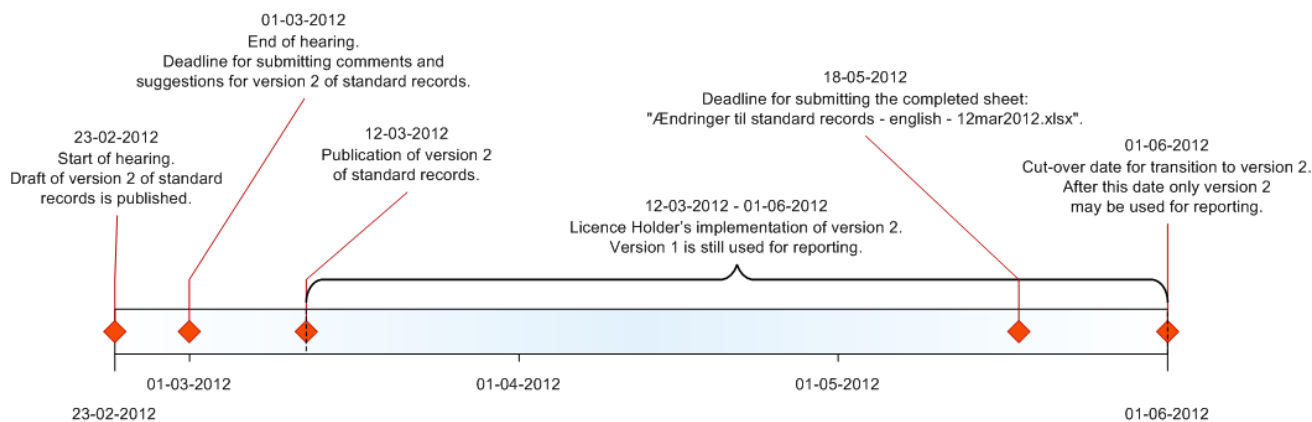
Documents

Ændringer til Standard Records – 12mar2012.xls

Ændringer til Standard Records – english – 12mar2012.xls

There will be a fixed cut-over date for transition between version 1 and version 2 of standard records. The cut-over date is June 1st, 2012. Cut-over should be handled in the following way: Zip-files packed with a token collected on May 31st or earlier, must contain standard records of version 1, and zip-files packed with a token collected on June 1st or later, must contain standard records of version 2.

Timeline for hearing, publication of version 2 and cut-over to version 2 of Standard Records:



Neglecting to submit standard records from June 1st, 2012, in the new version may affect your licence and whether or not your one-year licence will be changed to a five-year licence.

1.2.1. Details on cutover between v1 and v2

Below are details regarding transition from v1 to v2 of standard records.

- Cut-over
 - On June 1st at 00:00 UTC (i.e. June 1st at 02:00 CET).
 - Any tamper-token extracted before the cutover will seal data on v1 data model.
 - Any tamper-token extracted after the cutover will seal data on v2 data model.
- Transition period
 - It is not acceptable to submit v2 data before the cutover.
 - It is acceptable to extract tamper tokens before the cutover and keep them open for use after the cutover to seal v1 data.
 - A zip-file may not contain both v1 and v2 data. It's either v1 or v2 data.
 - A zip-file may not contain more than 1 day's data.
 - v1 data tamper tokens can remain open even after they're officially expired after 24 hours.
 - v1 tamper tokens should however be:
 - Ideally closed by June 3rd
 - At the very latest closed by June 8th.
 - v1 data recorded after June 1st will be stored in the May 31st folder. That folder may for example contain data for May 31st and June 1st, whereas the June 1st folder may stay empty.
 - Rollbacks from v2 model to v1 model are not acceptable. Once an operator (and all its suppliers) has switched to v2, even if data is faulty, it should still be reported as v2 and fixed as soon as possible.

- End-of day reports
 - May 31st is the final end-of-day report in v1 data model, covering data from May 30th.
 - June 1st is the first end-of-day report in v2 data model (underlying data still stored in v1 though). It will cover a period of 26 hours from May 31st 00:00 CET until June 1st 02:00 CET.
 - From June 2nd onwards, end-of-day reports will cover 24-hour periods starting at 00:00 UTC using the v2 data model (underlying data also stored in v2).

2. Standard Records

Licence Holder must report data in a data format called Standard Records specified by the Danish Gambling Authority. Standard Records are xml structures which are constructed based on elements which are explained in a conceptual model. The conceptual model is described in more detail in section 2.1. The specification of the exact xml-structures are explained in section 2.2 with reference to the XSD schemas. Standard records consists of a total of 18 different structures, which must be reported for the different game types. The 18 different structures and when they must be reported are explained in sections 2.2.1 to 2.2.8.

2.1. Conceptual model

The conceptual model should be considered a meta-model of the data elements which are used in Standard Records. The purpose is to explain the logical connections and to explain the data elements and their data types. The different data elements are used in several different Standard Records. The model is not a database model. The Danish Gambling Authority should not have access to a database, but collects all data as Standard Records packed as zip-files on the Licence Holder's SAFE.

The conceptual model is found as a separate document on the DGA's website.

Documents and files

Begrebsmodel Standard Records – 12mar2012.pdf
Begrebsmodel Standard Records – english – 12mar2012.pdf

2.2. XSD schemas for Standard Records

The DGA has specified a XSD schema for each of the 18 Standard Record types. All schemas are found in the zip-file mentioned in the table below. The zip-file contains the following folders:

- view: contains the XSD files which are specifying each type of Standard Record
- types: contains the different data types, which are used in the XSD schemas
- class: containa the XSD classes which are containing all the elements (also refered to as attributes), which are used in Standard Records

Licence Holder must follow the schemas strictly. This means that Licence Holder must report all data in the specified schemas. It is a requirement that Licence Holder report data which are schema valid and all relevant data must be reported. If Licence Holder is in doubt of which XSD schema to use for a specific game, the Licence Holder must contact the DGA who can inform of the right schema to use.

Documents and files

Standard Record XSD schemas – 24may2012.zip

The DGA has generated example files for illustration of Standard Records. The example files are generated automatically and illustrate syntax and data types. The actual content of each attribute is described in the conceptual model.

Documents and files
Standard Record examples – 24may2012.zip

2.2.1. Schema for fixed odds games

Licence Holder must report fixed odds games in two separate structures: one for reporting of transactions (stakes and cancellations) and one for reporting of prizes.

Keys that bind structures together for fixed odds games:

- 1) SpilTransaktionIdentifikation connects a gamblers stake with a cancellation, which both are reported in a FastOddsTransaktionStruktur.
- 2) SpilTransaktionIdentifikation connects a gamblers stake with a prize. Stakes are reported in a FastOddsTransaktionStruktur and prizes are reported in a FastOddsSlutStruktur.

Name of structure	Description
FastOddsTransaktionStruktur	The structure should be used for storing transactions for fixed odds games. Should be stored minimum every 5 minutes if there have been any transactions.
FastOddsSlutStruktur	The structure should be used for storing data directly after a fixed odds game is settled.

2.2.2. Schemas for casino games

Licence Holder must report casino games in one type of structure. The Licence Holder may choose between two types of structures – one structure where data is reported per session and one structure where data is reported per draw (spin). Further descriptions of the difference between session and draw can be found in the conceptual model. Both stakes and prizes are reported in the same structure.

Keys that bind structures together for casino games:

- 1) SpilTransaktionIdentifikation connects a gamblers stake with a cancellation, which are both reported in a KasinospilPrSessionStruktur/KasinoSpilPrTrækStruktur.

Name of structure	Description
KasinospilPrSessionStruktur	Structure for reporting casino games per session. Every file can contain several sessions but should be stored minimum every 5 minutes if there have been any transactions. The file should only contain data for ended sessions. The gambling provider should provide either KasinoSpilPrSessionStruktur or Kasi-

	nospilPrTrækStruktur.
KasinoSpilPrTrækStruktur	Structure for reporting casino games per draw. Every file can contain several draws but should be stored minimum every 5 minutes if there have been any transactions. The gambling provider should provide either KasinoSpilPrSessionStruktur or Kasi-nospilPrTrækStruktur.

2.2.3. Schemas for poker cash games

The Licence Holder must report data from poker cash games in one structure. The Licence Holder may choose between two types of structures – one structure where data is reported per session and one structure where data is reported per hand. Further descriptions of the difference between session and hand can be found in the conceptual model. Both stakes and prizes are reported in the same structure.

Keys that bind structures together for poker cash games:

- 1) SpilTransaktionIdentifikation connects a gamblers stake with a cancellation, which are both reported in a PokerCashGamePrSessionStruktur/PokerCashGamePrHåndStruktur.

Name of structure	Description
PokerCashGamePrHåndStruktur	Structure for reporting poker cash games per hand. Every file can contain several hands but should be stored minimum every 5 minutes if there have been any transactions. The gambling provider should provide either PokerCashGamePrSessionStruktur or PokerCashGamePrHåndStruktur.
PokerCashGamePrSessionStruktur	Structure for reporting poker cash games per session. Every file can contain several sessions but should be stored minimum every 5 minutes if there have been any transactions. The gambling provider should provide either PokerCashGamePrSessionStruktur or PokerCashGamePrHåndStruktur.

2.2.4. Schemas for poker tournaments

The Licence Holder must report data from poker tournaments in three different structures: one for reporting the start of a poker tournament (PokerTurneringStartStruktur), one for reporting transactions such as stakes and cancellations (PokerTurneringTransaktionStruktur) and one holding summarized data from the poker tournament (PokerTurneringSlutStruktur).

Keys that bind structures together for a poker tournament:

- 1) SpilProduktIdentifikation connects the StartStruktur, TransaktionStruktur and SlutStruktur for a poker tournament.
- 2) SpilTransaktionIdentifikation connects a gamblers stake with a cancellation which both are reported in a PokerTurneringTransaktionStruktur.

Name of structure	Description
PokerTurneringStartStruktur	The structure should be used for storing data directly after the gambling provider set up a poker tournament.
PokerTurneringTransaktionStruktur	The structure should be used for storing transactions in a poker tournament. Should be stored minimum every 5 minutes if there have been any transactions.
PokerTurneringSlutStruktur	The structure should be used for storing data directly after a poker tournament is over.

2.2.5. Schemas for pool games

The Licence Holder must report data from pool games in four different structures: one for reporting the start of a pool game (PuljespilStartStruktur), one for reporting transactions such as stakes and cancellations (PuljespilTransaktionStruktur), one holding summarized data after the game has closed for stakes and cancellations (PuljespilEndOfGameStruktur) and one holding summarized data for the pool game including the results (PuljespilSlutStruktur).

Keys that bind structures for pool games together:

- 1) SpilProduktIdentifikation connects StartStruktur, TransaktionStruktur, EndOfGameStruktur and SlutStruktur for a pool game.
- 2) SpilTransaktionIdentifikation connects a gamblers stake with a cancellation, which are both reported in a PuljespilTransaktionStruktur.

Structure name	Description
PuljespilStartStruktur	The structure should be used directly after a gambling provider set up a pool game.
PuljespilTransaktionStruktur	The structure should be used for reporting all players gaming transactions on a pool game. Should be stored minimum every 5 minutes if there have been any transactions.
PuljespilEndOfGameStruktur	The structure should be used for storing data at end-of-game in a pool game. Should be stored after the closing of stakes and cancellations, but before the first match begins.
PuljespilSlutStruktur	The structure should be used directly after a pool game has ended.

2.2.6. Schemas for manager games

The Licence Holder must report data from manager games (such as fantasy football) in three different structures: one for reporting the start of a manager game (ManagerSpilStartStruktur), one for reporting transactions such as stakes and cancellations (ManagerSpilTransaktionStruktur) and one for summarized data from the manager game (ManagerSpilSlutStruktur).

Keys that bind structures together for a manager game:

- 1) SpilProduktIdentifikation connects a StartStruktur, TransaktionStruktur and a SlutStruktur for a manager game.
- 2) SpilTransaktionIdentifikation connects a gamblers stake with a cancellation, which both are reported in a ManagerSpilTransaktionStruktur.

Structure name	Description
ManagerSpilStartStruktur	The structure should be used directly after a gambling provider sets up a manager game.
ManagerSpilTransaktionStruktur	The structure should be used for reporting all players gaming transactions on a manager game. Should be stored minimum every 5 minutes if there have been any transactions.
ManagerspilSlutStruktur	The structure should be used directly after a manager game has ended.

2.2.7. Schema for the triggering of a jackpot

The Licence Holder must report data from the triggering of jackpots in one structure (JackpotUdløsningStruktur).

Keys for JackpotUdløsningStruktur:

- 1) JackpotIdentifikation and SpillerInformationIdentifikation connects transaction structures and JackpotUdløsningStruktur. From these two values it must be possible for the DGA to verify that the gambler has put stakes into the jackpot pool.

Name of structure	Description
JackpotUdløsningStruktur	Structure for reporting a triggered jackpot which the gambling provider has taken part of. The data structure should be stored every time a jackpot has been triggered, also if the game is being played in a network and the jackpot has been triggered by a player of one of the other gambling providers.

2.2.8. Schema for End of Day reporting

This scheme contains summarized data for the Licence Holder. The report contains summarized data for the following categories under SpilKategoriNavn: Fastoddsspil, FastoddsspilBetexchange, PokerCashGame, KasinospilSinglePlayer and KasinospilMultiPlayer.

The End of Day report must not be generated for the following categories under SpilKategoriNavn: Puljespil, Managerspil og PokerTurnering. See the conceptual model in section 2.1 for definition of SpilKategoriNavn.

Licence Holder must generate one report per currency played by Danish gamblers during the day.

Name of structure	Description
EndOfDayRapportStruktur	The structure should be used for storing status data once per day. A day is defined after UTC time and is from 00:00 to 23:59. The file should at the latest be stored at 04:00 for the previous day.

The fields of the End of Day report should be calculated as described below:

SpilKategoriNavn	Attributes in the End of Day report	Description of calculations
Fastoddsspil		
	EndOfDayRapportAntalSpil	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Antal FastOddsTransaktionStruktur.SpilTransaktionIdentifikation without cancellation (o FastOddsTransaktionStruktur.SpilAnnullering = 0).</p> <p>B = Antal FastOddsTransaktionStruktur.SpilTransaktionIdentifikation with FastOddsTransaktionStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = FastOddsTransaktionStruktur.SpilKøbDatoTid.</p> <p>B: EndOfDayRapportDato = FastOddsTransaktionStruktur.SpilAnnulleringDatoTid.</p>
	EndOfDayRapportIndskudSpil	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Sum FastOddsTransaktionStruktur.SpilIndskud without cancellation (or FastOddsTransaktionStruktur.SpilAnnullering = 0).</p> <p>B = Sum FastOddsTransaktionStruktur.SpilIndskud with FastOddsTransaktionStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = FastOddsTransaktionStruktur.SpilKøbDatoTid.</p> <p>B: EndOfDayRapportDato = FastOddsTransaktionStruktur.SpilAnnulleringDatoTid.</p>
	EndOfDayRapportIndskudJackpot	Always reported as 0 (zero) for fixed odds games since jackpot is normally not used in connection with fixed odds games.

	EndOfDayRapportGevinster	<p><u>Result = A</u></p> <p>Calculated as:</p> <p>A = Sum FastOddsSlutStruktur.SpilGevinst.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = FastOddsSlutStruktur.SpilFaktiskSlutDatoTid.</p> <p>See section 2.3.3 for handling of resettlements.</p>
	EndOfDayRapportKommissionRake	<p><u>Result = A</u></p> <p>Calculated as:</p> <p>A = Sum FastOddsSlutStruktur.SpilKommission.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = FastOddsSlutStruktur.SpilFaktiskSlutDatoTid.</p> <p>See section 2.3.3 for handling of resettlements.</p>
FastoddsspilBetexchange		
	EndOfDayRapportAntalSpil	Same as Fastoddsspil.
	EndOfDayRapportIndskudSpil	Same as Fastoddsspil.
	EndOfDayRapportIndskudJackpot	Same as Fastoddsspil.
	EndOfDayRapportGevinster	Same as Fastoddsspil.
	EndOfDayRapportKommissionRake	Same as Fastoddsspil.
PokerCashGame (reporting using PokerCashGamePrHåndStruktur)		
	EndOfDayRapportAntalSpil	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Antal PokerCashGamePrHåndStruktur.SpilTransaktionIdentifikation without cancellation (or PokerCashGamePrHåndStruktur.SpilAnnullering = 0).</p> <p>B = Antal PokerCashGamePrHåndStruktur.SpilTransaktionIdentifikation with PokerCashGamePrHåndStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = PokerCashGamePrHåndStruktur.SpilFaktiskSlutDatoTid.</p> <p>B: EndOfDayRapportDato = PokerCashGamePrHåndStruktur.SpilAnnulleringDatoTid.</p>
	EndOfDayRapportIndskudSpil	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Sum PokerCashGamePrHåndStruktur.PokerHåndIndskudSpil without cancellation (or PokerCashGamePrHåndStruktur.SpilAnnullering = 0).</p> <p>B = Sum PokerCashGamePrHåndStruktur.</p>

		<p>tur.PokerHåndIndskudSpil with PokerCashGamePrHåndStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = PokerCashGamePrHåndStruktur.SpilFaktiskSlutDatoTid.</p> <p>B: EndOfDayRapportDato = PokerCashGamePrHåndStruktur.SpilAnnulleringDatoTid.</p>
	EndOfDayRapportIndskudJackpot	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Sum PokerCashGamePrHåndStruktur.PokerHåndIndskudJackpot without cancellation (or PokerCashGamePrHåndStruktur.SpilAnnullering = 0).</p> <p>B = Sum PokerCashGamePrHåndStruktur.PokerHåndIndskudJackpot with PokerCashGamePrHåndStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = PokerCashGamePrHåndStruktur.SpilFaktiskSlutDatoTid.</p> <p>B: EndOfDayRapportDato = PokerCashGamePrHåndStruktur.SpilAnnulleringDatoTid.</p>
	EndOfDayRapportGevinster	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Sum PokerCashGamePrHåndStruktur.PokerHåndGevinstSpil without cancellation (or PokerCashGamePrHåndStruktur.SpilAnnullering = 0).</p> <p>B = Sum PokerCashGamePrHåndStruktur.PokerHåndGevinstSpil with PokerCashGamePrHåndStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = PokerCashGamePrHåndStruktur.SpilFaktiskSlutDatoTid.</p> <p>B: EndOfDayRapportDato = PokerCashGamePrHåndStruktur.SpilAnnulleringDatoTid.</p>
	EndOfDayRapportKommissionRake	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Sum PokerCashGamePrHåndStruktur.PokerHåndRake without cancellation (or PokerCashGamePrHåndStruktur.SpilAnnullering = 0).</p> <p>B = Sum PokerCashGamePrHåndStruktur.PokerHåndRake with PokerCashGamePrHåndStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = PokerCashGamePrHåndStruktur.SpilFaktiskSlutDatoTid.</p> <p>B: EndOfDayRapportDato = PokerCashGamePrHåndStruktur.SpilAnnulleringDatoTid.</p>
PokerCashGame (reporting using Poker-		

CashGamePrSessionStruktur)		
	EndOfDayRapportAntalSpil	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Antal PokerCashGamePrSessionStruktur.PokerSessionAntalHænder without cancellation (or PokerCashGamePrSessionStruktur.SpilAnnullering = 0).</p> <p>B = Antal PokerCashGamePrSessionStruktur.PokerSessionAntalHænder with PokerCashGamePrSessionStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = PokerCashGamePrSessionStruktur.SpilFaktiskSlutDatoTid.</p> <p>B: EndOfDayRapportDato = PokerCashGamePrSessionStruktur.SpilAnnulleringDatoTid.</p>
	EndOfDayRapportIndskudSpil	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Sum PokerCashGamePrSessionStruktur.PokerSessionIndskudSpil without cancellation (or PokerCashGamePrSessionStruktur.SpilAnnullering = 0).</p> <p>B = Sum PokerCashGamePrSessionStruktur.PokerSessionIndskudSpil with PokerCashGamePrSessionStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = PokerCashGamePrSessionStruktur.SpilFaktiskSlutDatoTid.</p> <p>B: EndOfDayRapportDato = PokerCashGamePrSessionStruktur.SpilAnnulleringDatoTid.</p>
	EndOfDayRapportIndskudJackpot	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Sum PokerCashGamePrSessionStruktur.PokerSessionIndskudJackpot without cancellation (or PokerCashGamePrSessionStruktur.SpilAnnullering = 0).</p> <p>B = Sum PokerCashGamePrSessionStruktur.PokerSessionIndskudJackpot with PokerCashGamePrSessionStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = PokerCashGamePrSessionStruktur.SpilFaktiskSlutDatoTid.</p> <p>B: EndOfDayRapportDato = PokerCashGamePrSessionStruktur.SpilAnnulleringDatoTid.</p>
	EndOfDayRapportGevinster	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Sum PokerCashGamePrSessionStruktur.PokerSessionGevinstSpil without cancellation (or PokerCashGamePrSessionStruktur.SpilAnnullering = 0).</p>

		<p>B = Sum PokerCashGamePrSessionStruktur.PokerSessionGevinstSpil with PokerCashGamePrSessionStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = PokerCashGamePrSessionStruktur.SpilFaktiskSlutDatoTid.</p> <p>B: EndOfDayRapportDato = PokerCashGamePrSessionStruktur.SpilAnnulleringDatoTid.</p>
	EndOfDayRapportKommissionRake	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Sum PokerCashGamePrSessionStruktur.PokerSessionRake without cancellation (or PokerCashGamePrSessionStruktur.SpilAnnullering = 0).</p> <p>B = Sum PokerCashGamePrSessionStruktur.PokerSessionRake with PokerCashGamePrSessionStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = PokerCashGamePrSessionStruktur.SpilFaktiskSlutDatoTid.</p> <p>B: EndOfDayRapportDato = PokerCashGamePrSessionStruktur.SpilAnnulleringDatoTid.</p>
KasinospilSinglePlayer (reporting using KasinospilPrTraekStruktur)		
	EndOfDayRapportAntalSpil	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Antal KasinospilPrTraekStruktur.SpilTransaktionIdentifikation without cancellation (or KasinospilPrTraekStruktur.SpilAnnullering = 0).</p> <p>B = Antal KasinospilPrTraekStruktur.SpilTransaktionIdentifikation with KasinospilPrTraekStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = KasinospilPrTraekStruktur.SpilFaktiskSlutDatoTid.</p> <p>B: EndOfDayRapportDato = KasinospilPrTraekStruktur.SpilAnnulleringDatoTid.</p>
	EndOfDayRapportIndskudSpil	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Sum KasinospilPrTraekStruktur.KasinoTraekIndskudSpil without cancellation (or KasinospilPrTraekStruktur.SpilAnnullering = 0).</p> <p>B = Sum KasinospilPrTraekStruktur.KasinoTraekIndskudSpil with KasinospilPrTraekStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = KasinospilPrTraekStruktur.SpilFaktiskSlutDatoTid.</p>

		B: EndOfDayRapportDato = KasinospilPrTraekStruktur.SpilAnnulleringDatoTid.
	EndOfDayRapportIndskudJackpot	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Sum KasinospilPrTraekStruktur.KasinoTrækIndskudJackpot without cancellation (or KasinospilPrTraekStruktur.SpilAnnullering = 0).</p> <p>B = Sum KasinospilPrTraekStruktur.KasinoTrækIndskudJackpot with KasinospilPrTraekStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = KasinospilPrTraekStruktur.SpilFaktiskSlutDatoTid.</p> <p>B: EndOfDayRapportDato = KasinospilPrTraekStruktur.SpilAnnulleringDatoTid.</p>
	EndOfDayRapportGevinster	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Sum KasinospilPrTraekStruktur.KasinoTrækGevinstSpil without cancellation (or KasinospilPrTraekStruktur.SpilAnnullering = 0).</p> <p>B = Sum KasinospilPrTraekStruktur.KasinoTrækGevinstSpil with KasinospilPrTraekStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = KasinospilPrTraekStruktur.SpilFaktiskSlutDatoTid.</p> <p>B: EndOfDayRapportDato = KasinospilPrTraekStruktur.SpilAnnulleringDatoTid.</p>
	EndOfDayRapportKommissionRake	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Sum KasinospilPrTraekStruktur.KasinoTrækKommission without cancellation (or KasinospilPrTraekStruktur.SpilAnnullering = 0).</p> <p>B = Sum KasinospilPrTraekStruktur.KasinoTrækKommission with KasinospilPrTraekStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = KasinospilPrTraekStruktur.SpilFaktiskSlutDatoTid.</p> <p>B: EndOfDayRapportDato = KasinospilPrTraekStruktur.SpilAnnulleringDatoTid.</p>
KasinospilMultiPlayer (reporting using KasinospilPrTraekStruktur)		
	EndOfDayRapportAntalSpil	Same as KasinospilSinglePlayer reported using KasinospilPrTraekStruktur.
	EndOfDayRapportIndskudSpil	Same as KasinospilSinglePlayer reported using KasinospilPrTraekStruktur.
	EndOfDayRapportIndskudJackpot	Same as KasinospilSinglePlayer reported using Kasi-

		nospilPrTraekStruktur.
	EndOfDayRapportGevinster	Same as KasinospilSinglePlayer reported using KasinospilPrTraekStruktur.
	EndOfDayRapportKommissionRake	Same as KasinospilSinglePlayer reported using KasinospilPrTraekStruktur.
KasinospilSinglePlayer (reporting using KasinospilPrSessionStruktur)		
	EndOfDayRapportAntalSpil	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Antal KasinospilPrSessionStruktur.KasinospilAntalTraek without cancellation (or KasinospilPrSessionStruktur.SpilAnnullering = 0).</p> <p>B = Antal KasinospilPrSessionStruktur.KasinospilAntalTraek with KasinospilPrSessionStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = KasinospilPrSessionStruktur.SpilFaktiskSlutDatoTid.</p> <p>B: EndOfDayRapportDato = KasinospilPrSessionStruktur.SpilAnnulleringDatoTid.</p>
	EndOfDayRapportIndskudSpil	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Sum KasinospilPrSessionStruktur.KasinospilIndskudSpil without cancellation (or KasinospilPrTraekStruktur.SpilAnnullering = 0).</p> <p>B = Sum KasinospilPrSessionStruktur.KasinospilIndskudSpil with KasinospilPrSessionStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = KasinospilPrSessionStruktur.SpilFaktiskSlutDatoTid.</p> <p>B: EndOfDayRapportDato = KasinospilPrSessionStruktur.SpilAnnulleringDatoTid.</p>
	EndOfDayRapportIndskudJackpot	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Sum KasinospilPrSessionStruktur.KasinospilIndskudJackpot without cancellation (or KasinospilPrSessionStruktur.SpilAnnullering = 0).</p> <p>B = Sum KasinospilPrSessionStruktur.KasinospilIndskudJackpot with KasinospilPrSessionStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = KasinospilPrSessionStruktur.SpilFaktiskSlutDatoTid.</p> <p>B: EndOfDayRapportDato = KasinospilPrSessionStruktur.SpilAnnulleringDatoTid.</p>
	EndOfDayRapportGevinster	<p><u>Result = A - B</u></p> <p>Calculated as:</p>

		<p>A = Sum KasinospilPrSessionStruktur.KasinospilGevinstSpil without cancellation (or KasinospilPrSessionStruktur.SpilAnnullering = 0).</p> <p>B = Sum KasinospilPrSessionStruktur.KasinospilGevinstSpil with KasinospilPrSessionStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = KasinospilPrSessionStruktur.SpilFaktiskSlutDatoTid.</p> <p>B: EndOfDayRapportDato = KasinospilPrSessionStruktur.SpilAnnulleringDatoTid.</p>
	EndOfDayRapportKommissionRake	<p><u>Result = A - B</u></p> <p>Calculated as:</p> <p>A = Sum KasinospilPrSessionStruktur.KasinospilKommission without cancellation (or KasinospilPrSessionStruktur.SpilAnnullering = 0).</p> <p>B = Sum KasinospilPrSessionStruktur.KasinospilKommission with KasinospilPrSessionStruktur.SpilAnnullering = 1.</p> <p>Data is chosen based on the following rules for dates:</p> <p>A: EndOfDayRapportDato = KasinospilPrSessionStruktur.SpilFaktiskSlutDatoTid.</p> <p>B: EndOfDayRapportDato = KasinospilPrSessionStruktur.SpilAnnulleringDatoTid.</p>
KasinospilMultiPlayer (reported using KasinospilPrSessionStruktur)		
	EndOfDayRapportAntalSpil	Same as KasinospilSinglePlayer reported using KasinospilPrSessionStruktur.
	EndOfDayRapportIndskudSpil	Same as KasinospilSinglePlayer reported using KasinospilPrSessionStruktur.
	EndOfDayRapportIndskudJackpot	Same as KasinospilSinglePlayer reported using KasinospilPrSessionStruktur.
	EndOfDayRapportGevinster	Same as KasinospilSinglePlayer reported using KasinospilPrSessionStruktur.
	EndOfDayRapportKommissionRake	Same as KasinospilSinglePlayer reported using KasinospilPrSessionStruktur.
Puljespil	<i>Pool game may not be reported in the End of Day report.</i>	
PokerTurnering	<i>Poker pournamenets may not be reported in the End of Day report.</i>	
Managerspil	<i>Manager games may not be reported in the End of Day report.</i>	

2.3. Error handling

This section describes how errors in data must be corrected. The DGA specifies four types of errors: cancellations, bet-resettlements, replacement data and other error types. In this section it is described how Licence Holder should correct each type of error.

As a general requirements, all data must be placed on SAFE and data on SAFE must be kept as described in the document Requirements for Licence Holder. This means that data must be kept on SAFE in the required period also even though it contains errors. No data may be deleted from SAFE before the required period for keeping data on SAFE has been exceeded.

Below is described the different ways of correcting errors. Not that the rules mentioned below should be followed strictly. This means that it is not possible to use the attribute for replacement data for handling cancellations or bet-resettlement.

Type	Description	Applies to the following Standard Records
Cancellation	Changes to Transaktion-, Træk-, and SessionStruktur are handled as cancellations. Changes must be performed by sending a cancellation followed by a new transaction. See further details in section 2.3.1.	FastOddsTransaktionStruktur KasinospilPrSessionStruktur KasinoSpilPrTrækStruktur ManagerSpilTransaktionStruktur PokerCashGamePrHåndStruktur PokerCashGamePrSessionStruktur PokerTurneringTransaktionStruktur PuljespilTransaktionStruktur
Bet-resettlement	Changes for FastOddsSlutStruktur is handled as bet-resettlement. This means that the Licence Holder must send another FastOddsSlutStruktur with the difference in prize. See further details in section 2.3.3.	FastOddsSlutStruktur
Replacement data	Changes in Start- and SlutStruktur for manager games, poker tournaments, pool games, End-OfGame of pool games and End-OfDay must be handled using the attribute for replacement data. See further details in section 2.3.5.	PuljespilStartStruktur ManagerSpilStartStruktur PokerTurneringStartStruktur EndOfDayRapportStruktur PuljespilEndOfGameStruktur PokerTurneringSlutStruktur PuljespilSlutStruktur ManagerspilSlutStruktur JackpotUdløsningStruktur
Other error types	Other types of errors are handled ad hoc in a dialog with the DGA. These types of errors could be corrupt files, schema validation errors and the like.	Applies to all structures.

	See further details in section 2.3.7.	
--	---------------------------------------	--

2.3.1. Cancellations

Correction of stake transactions must always be handled as a cancellation and possibly followed by a new and correct stake transaction. If a new stake transaction is sent, the new transaction must have a new transaction id in the attribute SpilTransaktionIdentifikation. Licence Holder may not overwrite a stake transaction but only cancel it. Cancellation is performed by filling out the relevant transaction structure with exactly the same information as originally reported, but also setting the attribute SpilAnnullering to 1 and report a value for SpilAnnulleringDatoTid. The DGA will use the attribute SpilTransaktionIdentifikation as key to connect the stake transaction to the cancellation transaction. If Licence Holder by a mistake has reported other values in the cancellation transaction compared to the stake transaction (e.g. different amounts), then the DGA will use the value from the stake transaction and ignore the value in the cancellation transaction.

Cancellation may be used with the following structures:

- FastOddsTransaktionStruktur
- KasinospilPrSessionStruktur
- KasinoSpilPrTrækStruktur
- ManagerSpilTransaktionStruktur
- PokerCashGamePrHåndStruktur
- PokerCashGamePrSessionStruktur
- PokerTurneringTransaktionStruktur
- PuljespilTransaktionStruktur

2.3.2. Cancellations and the End of Day report

The End of Day report should not be re-generated if Licence Holder applies cancellations. If a stake transaction is reported on day 1 and a cancellation of the stake transaction is reported on day 2, then the stake transaction must be part of day 1 End of Day report and the cancellation must be part of day 2 End of Day report. This is also described in section 2.2.8 under **”Data is chosen based on the following rules for dates:”**.

2.3.3. Bet-resettlement

In the situation where a result in a fixed odds game has been incorrectly registered and the prizes therefore must be changed of one or several bets, then the Licence Holder must report the prize difference in a new FastOddsSlutStruktur. For example, if a gambler initially has won 100 DKK and this amount has been reported in a FastOddsSlutStruktur, but after the correction of the result, the gambler should not have a prize, the value of -100 DKK for the gambler should be reported in a new FastOddsSlutStruktur but with the same value for SpilTransaktionIdentifikation. The DGA will then use the attribute SpilTransaktionIdentifikation to connect the different prizes for the gambler on the game and summarize the prizes to calculate the total prize for the gambler on this game.

Bet-resettlement may be used with the following structures:

- FastOddsSlutStruktur

By using bet-resettlement, a new value of SpilFaktiskSlutDatoTid must be reported which matches the day and time where the bet-resettlement was performed.

2.3.4. Bet-resettlement and the End of Day report

The End of Day report should not be re-generated if Licence Holder uses bet-resettlement. If the Licence Holder first reports a FastOddsSlutStruktur on day 1 and another FastOddsSlutStruktur as bet-resettlement on day 2, then then first FastOddsSlutStruktur must be part of day 1 End of Day report and the other FastOddsSlutStruktur must be part of day 2 End of Day report. This is also described in section 2.2.8 under ”**Data is chosen based on the following rules for dates:**”.

2.3.5. Replacement data

The Licence Holder may replace specific data by using the attribute regarding replacement data (SpilFilErstatningIdentifikation). If the Licence Holder needs to replace data, it must also be reported through a specific webapplication for incident reporting. Also, Licence Holder must report why replacement data is necessary.

The field for replacement data may be used for the following structures:

- PuljespilStartStruktur
- ManagerSpilStartStruktur
- PokerTurneringStartStruktur
- EndOfDayRapportStruktur
- PuljespilEndOfGameStruktur
- PokerTurneringSlutStruktur
- PuljespilSlutStruktur
- ManagerspilSlutStruktur
- JackpotUdløsningStruktur

There are the following rules for using replacement data:

- 1) The original structure and the new structure must be of the same type. This means that a PuljespilStartStruktur may only be replaced by a PuljespilStartStruktur, and it will not be possible to replace a PuljespilStartStruktur with a ManagerSpilStartStruktur.

Replacement data must only be applied on specific occasions, since it requires manual approval from both the DGA and the Licence Holder.

2.3.6. Replacement data and the End of Day report

The End of Day report should not be regenerated if Licence Holder applies replacement data. This is since replacement data is used on structures for Puljespil, Managerspil, Pokerturning and Jackpotudløsning which are not reported in the End of Day report.

However, replacement data is applied to replace the End of Day report in other specific occasions.

2.3.7. Other error types

If the DGA or the Licence Holder discovers other error types in data, then the Licence Holder must correct the error and resend all zip-files and xml-files which contains the error. This also account for historic data.

To resend data the Licence Holder must retrieve a new token and pack data again. Licence Holder must report this to the DGA through the webbased incident reporting system, and may not delete data with errors from SAFE.

With respect to other error types, the DGA understands that there may be a short running-in period where Licence Holder may find and correct errors in the code that generates data. However, in the long run the amount of data errors will be a significant parameter in relation to the ongoing risk assesment of Licence Holders, specifically with respect to extension or renewal of licence.

2.3.8. Other error types and End of Day report

There may be a need for Licence Holder to generate the End of Day report if it is discovered that there are other error types in the reported data. The End of Day report should as a rule of thumb be re-generated and re-sent if there are errors in the data that are used for calculating the End of Day report. If the End of Day report is re-generated the Licence Holder must report this to the DGA.

2.4. Release process for Standard Records

The DGA will update Standard Records as needed. This could be in order to ensure high data quality, to extend the controls which the DGA performs or to implement suggestions from the Licence Holders.

Updates to Standard Records will be divided into major and minor releases.

2.4.1. Major releases

Major releases will be changes to Standard Records, where a larger effort from Licence Holder is expected. Major releases will not be backwards compatible and an effort from all Licence Holders will be necessary to adjust to the new release.

A major release will be announced beforehand on the DGA's website. Licence Holder will have good time for implementing the changes, where the time will depend on the number of changes. The DGA will communicate a cut-over date where the earlier version of Standard Records are no longer valid, and each Licence Holder must therefore use the new version of Standard Records from this date. It will not be possible for a Licence Holder both to report in a new and an old version of Standard Records at the same time.

Licence Holder must ensure that there are no data errors when the cut-over is performed, e.g. through a thorough test.

2.4.2. Minor releases

Minor releases are smaller changes to Standard Records, which are backwards compatible. This could be extension of data domains, addition of optional elements, etc.

Minor releases will not be announced beforehand and will be announced on the DGA's website from day to day. Minor releases will not necessarily affect all Licence Holders, but the Licence Holders for who the minor release is relevant must quickly adapt to the new version. A cut-over date may be set for specific Licence Holders for who the release is relevant, but it might not be a cut-over date which applies to all Licence Holders.

Similarly to major releases, Licence Holder must ensure that there are no data errors by the transition to a new minor release. However, the amount of testing will most likely be less than for a major release.

3. SAFE

SAFE is Licence Holder's data storage for Standard Records. More information on SAFE may be found in the document Requirements for Licence Holder on the DGA's website.

Typically, each Licence Holder will have one SAFE. However, the DGA may approve several SAFEs per Licence Holder. If the Licence Holder wish to report data in on multiple SAFEs, the Licence Holder must contact the DGA with the number of SAFEs and explanation. The DGA will provide one username for the Tamper Token system per SAFE.

In the following is described the required folder structure on SAFE as well as other SAFE configurations.

3.1. Folderstructure on SAFE

The Licence Holder must build the SAFE based on this structure:

Level 1: The outermost folder should be named "folderstruktur-spilssystem".

Level 2: Here is one folder named "Zip".

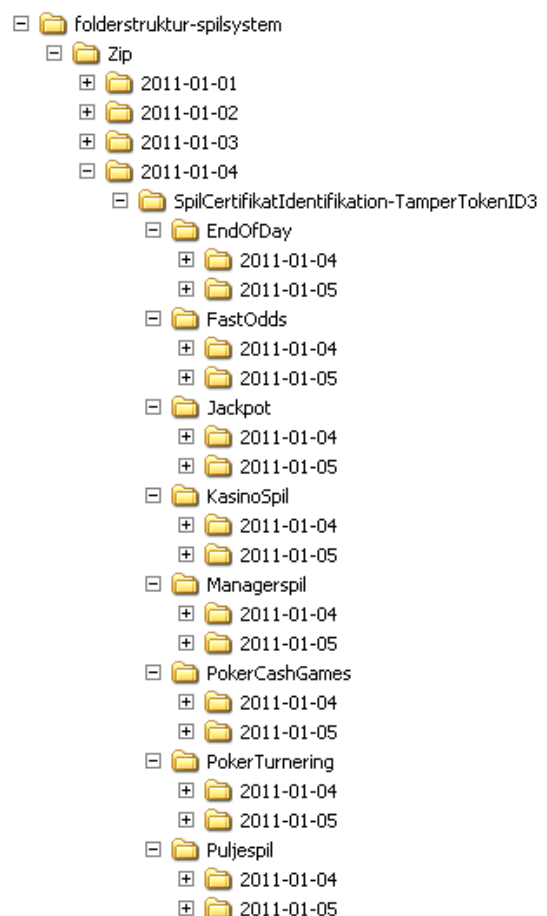
Level 3: Here are folders for each day, named after the date in the format YYYY-MM-DD.

Level 4: Here are a number of zip-files, where each is connected to one token. There are also folders for the tokens which are not yet closed. A folder which has not has been closed yet is named SpilCertifikatIdentifikation-TamperTokenID. The zip-file which contains the folder is named SpilCertifikatIdentifikation-TamperTokenID.zip.

Level 5: Here are the folders each zip-file contains. They are named: "EndOfDay", "FastOdds", "Jackpot", "KasinoSpil", "Managerspil", "PokerCashGames", "PokerTurnering" and "Puljespil".

Level 6: Here are folders for the relevant dates, named after the date in the format YYYY-MM-DD. Each Standard Record is placed on this level or level 7, and is placed in the folder that matches the time where the file is created.

Level 7: [Optional] There is a possibility for creating subfolders containing time intervals in the format HH.MMHH-MM.



The Standard Record must be placed in the folder with the date that represent the creation of the file.

An example of the folder structure is shown to the right.

3.2. SAFE configurations

For the DGA to find and get the right data on SAFE, certain configurations must be in place in relation to naming and packing of files.

3.2.1. Naming standards

All files on SAFE must follow the naming standard provided below. This applies both to Standard Record files and zip-files.

Standard record files must be named: SpilCertifikatIdentifikation-TamperTokenID-SequenceInToken.txt.

The zip-files must be named: SpilCertifikatIdentifikation-TamperTokenID.zip.

Here is:

SpilCertifikatIdentifikation:	Text string provided by the Danish Gambling Authority to the License Holder during the connection process.
TamperTokenID:	Identification of each tamper token which is received by calling the service operation TamperTokenHent in the service TamperTokenAnvend.
SequenceInToken:	A serial number, running from 1 and ended with E for "End" (1, 2, 3,...,E) to indicate the sequence of which each Standard Record is included in the MAC algorithm for each token. It is the License Holder's responsibility to build a mechanism for generation of the sequence.

Example:

If SpilCertifikatIdentifikation=12345, TamperTokenID=67890 and SequenceInToken=3 for a specific Standard Record file, the Standard Record file must have the following name *12345-67890-3.xml*. The zip-file which contains this Standard Record must have the file name *12345-67890.zip*.

3.2.2. Packing of files

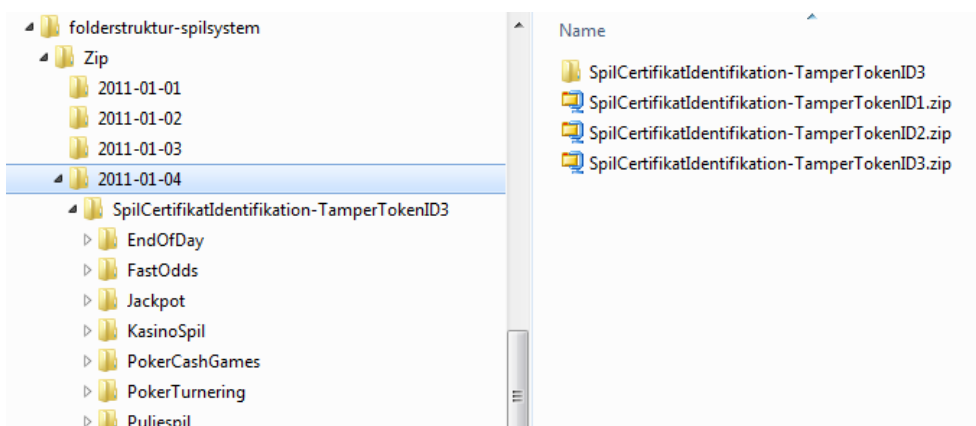
To save disk space for the License Holder and to simplify the transfer of files, the Standard Record files must be zipped in a continuous process when they appear on SAFE. How they should be zipped is explained in the following.

When the Standard Record files for a token arrives, the following should happen:

1. The MAC algorithm is run on each file as described earlier in the document.
2. The Standard Record is saved in the folder structure for the present token.
3. The Standard Record is appended the zip-file for the present token.

When the token is closed and all Standard Record files are added to the zip-file, the folder which matches the zip-file should be deleted. It is the License Holders task to build a mechanism to secure that these three steps are conducted correctly.

The following contains an example: On the figure below it is illustrated that the folder 2011-01-04 has two closed tokens which is associated to the zip-files: SpilCertifikatIdentifikation-TamperTokenID1.zip and SpilCertifikatIdentifikation-TamperTokenID2.zip, and one open token which is associated to SpilCertifikatIdentifikation-TamperTokenID3.zip. It can be seen that SpilCertifikatIdentifikation-TamperTokenID3.zip is open since there is both a zip-file and a folder structure with the same name. The Standard Records appearing continuously and is associated to token 3 should be saved in the folder SpilCertifikatIdentifikation-TamperTokenID3 and appended to SpilCertifikatIdentifikation-TamperTokenID3.zip. When token 3 is closed and all Standard Records are added to the zip file, the folder SpilCertifikatIdentifikation-TamperTokenID3 should be deleted.



3.2.3. Choice of date-folder for zip-file

As described in the section above, the zip-file must be placed under level 3 in the folderstructure on SAFE. Level 3 is folders with dates, and the zip-file must be placed under the correct date.

The zip-file must be placed under the issue-date which is valid for the token. The issue-date is found in the response from the serviceoperation TamperTokenHent and is the first 10 characters of the element TamperTokenUdstedelseDatoTid.

In the example below is given the value **2011-10-16T15:21:19.221+02:00** in the element **Tamper-TokenUdstedelseDatoTid**. The zip-file, which is build using this token, must therefore be found on SAFE under the path folderstruktur-spilssystem/Zip/2011-10-16/.

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <env:Body>
    <ns:TamperTokenAnvend_O xmlns:ns="http://skat.dk/begrebsmodel/2009/01/15/">
      <ns:Kontekst>
        <HovedOplysningerSvar xmlns="http://skat.dk/begrebsmodel/xml/schemas/kontekst/2007/05/31/">
          <TransaktionsID>895ffb40-9f4a-11e0-8264-0800200c9a66</TransaktionsID>
          <ServiceID>TamperTokenAnvendService</ServiceID>
          <TransaktionsTid>2011-06-25T18:41:30.054+01:00</TransaktionsTid>
        </HovedOplysningerSvar>
      </ns:Kontekst>
      <ns:TamperTokenHent_O>
        <ns:TamperTokenID>2149</ns:TamperTokenID>
        <ns:TamperTokenStartMAC>91c5e2c0e033e3b18fc66bfa43bb08d4</ns:TamperTokenStartMAC>
        <ns:TamperTokenUdstedelseDatoTid>2011-10-16T15:21:19.221+02:00</ns:TamperTokenUdstedelseDatoTid>
        <ns:TamperTokenPlanlagtLukketDatoTid>2011-10-17T15:21:19.221+02:00</ns:TamperTokenPlanlagtLukketDatoTid>
      </ns:TamperTokenHent_O>
    </ns:TamperTokenAnvend_O>
  </env:Body>
</env:Envelope>
```

3.2.4. UTC time for SAFE

SAFE must be configured for UTC time such that the timestamps of files and folders are given in UTC time.

4. Tamper Token and ROFUS

The purpose of this section is to provide instructions of service usage for the Tamper Token system and for the Problem Gambler Register (ROFUS).

In each subsection is described both general information which is relevant for both the Tamper Token and ROFUS services, but also the specific information which is relevant for either Tamper Token or ROFUS. Furthermore, it is described how the Licence Holder may test parts of the solution.

4.1. Tamper Token: Webservice description, WSDL- and XSD-files

The service description for the webservices which must be used on the Tamper Token system is described in the documents below.

Documents and files

TamperToken service – 03mar2011.pdf
TamperToken service – english – 03mar2011.pdf

WSDL-file and XSD-files are found in the file mentioned below.

Documents and files

Tamper Token service – xml – 03mar2011.zip
--

4.2. ROFUS: Webservice descriptions, WSDL- and XSD-files

The service description for the webservices which must be used on the ROFUS system is described in the document below.

Documents and files

ROFUS services – English – 01sep2011.pdf
--

WSDL-file and XSD-files are found in the file mentioned below.

Documents and files

ROFUS services – xml – 01sep2011.zip

4.3. General technical configurations

In the following is described general technical configurations for Tamper Token and ROFUS including handling of header information and error messages.

4.3.1. Header information and error handling

Header and error information is handled identically for Tamper Token and ROFUS services. The purpose of including header information is to make it possible to follow request and response for a service call and to be able to report errors.

The header information is inserted in an any-element on each service and must comply with the format specified in the XSD-files for header information, which are found on the Gambling Authority's website.

Documents and files

Header information xml - 07dec2010.zip

The following header information should be stated in a service request made by the License Holder:

- **TransaktionsID:** License Holder must generate a unique transaction id for the service call. The Gambling Authority recommends to follow the standard Universally Unique Identifier (UUID), where the id consists of 32 hexa decimals represented in 5 groups separated by dashes on the form 8-4-4-4-12. E.g.: 07B2A963-26C4-47E0-B517-C7059A598DA3
- **TransaktionsTid:** The time of transaction. The time must be stated on the form YYYY-MM-DDThh:mm:ss.sTZD, where YYYY is year, MM is month, DD is day, hh is hours, mm is minutes, ss is seconds, s is one or more digits for seconds, and TZD is the time zone represented as Z or +hh:mm or -hh:mm. E.g.: 2010-12-07T09:33:51.249+01:00.

The following header information is always stated in a service response:

- **TransaktionsID:** Same as above.
- **TransaktionsTid:** Same as above.
- **ServiceID:** The name of the called service.

The following header information is also returned in a service response, but is optional and is only returned when necessary.

- **Fejl:** Errors are reported when a request is not completed as expected.
 - **FejlNummer:** Id-number for the error.
 - **FejlTekst:** Description in text of the error.
 - **Identifikation:** Text-code for the error.
 - **ServiceID:** Same as above.
- **Advis:** Notifications are messages which are not errors. It could be a message explaining that the service call has been executed as expected.
 - **AdvisNummer:** Id-number for the notification.
 - **AdvisTekst:** Description in text of the notification.
 - **Identifikation:** Text-code for the notification.
 - **ServiceID:** Same as above.

Example, service-request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://services.lur.skat.dk">
  <soapenv:Header/>
```

```

<soapenv:Body>
  <ser:GamblerCreateRequest>
    <Kontekst>
      <ns1:HovedOplysninger xmlns:ns1="http://skat.dk/begrebsmodel/xml/schemas/kontekst/2007/05/31/">
        <ns1:TransaktionsID>07B2A963-26C4-47E0-B517-C7059A598DA3</ns1:TransaktionsID>
        <ns1:TransaktionsTid>2010-12-07T13:23:57.435+01:00</ns1:TransaktionsTid>
      </ns1:HovedOplysninger>
    </Kontekst>
    <Actor>Spillemyndigheden</Actor>
    <PersonInformation>
      <PersonCPRNumber>111111-3333</PersonCPRNumber>
    </PersonInformation>
    <ExclusionStart>
      <GameExclusionStartDate>2010-12-07</GameExclusionStartDate>
      <GameExclusionStartTime>13:23:57</GameExclusionStartTime>
    </ExclusionStart>
  </ser:GamblerCreateRequest>
</soapenv:Body>
</soapenv:Envelope>

```

Example, service-response:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:GamblerCreateResponse xmlns:ns3="http://services.lur.skat.dk"
      xmlns:ns2="http://skat.dk/begrebsmodel/xml/schemas/kontekst/2007/05/31/">
      <Kontekst>
        <ns2:HovedOplysningerSvar xmlns="http://skat.dk/begrebsmodel/xml/schemas/kontekst/2007/05/31/">
          <TransaktionsID>07B2A963-26C4-47E0-B517-C7059A598DA3</TransaktionsID>
          <ServiceID>LUR_SPILLER_OPRET</ServiceID>
          <TransaktionsTid>2010-12-07T13:44:43.510+01:00</TransaktionsTid>
          <SvarReaktion>
            <Advis>
              <AdvisNummer>7</AdvisNummer>
              <AdvisTekst>The person is registered as a gambler in the Ludomaniregister</AdvisTekst>
              <Identifikation>
                <IDText xmlns=""
          xmlns:ns5="http://skat.dk/begrebsmodel/xml/schemas/kontekst/2007/05/31/">LUR:SpillerIndberet.GamblerISReported</IDText>
              </Identifikation>
            </Advis>
          </SvarReaktion>
        </ns2:HovedOplysningerSvar>
      </Kontekst>
    </ns3:GamblerCreateResponse>
  </S:Body>
</S:Envelope>

```

4.3.2. Service call – creation of http(s) connection and receiving data

The Danish Gambling Authority has developed an example of service calls to help the License Holder's task of calling the webserivces. The example shows how one can (in Java and .Net) get webservice descriptions and call services by the use of http basic access authentication. Furthermore is shown how data can be received from the service. The service GamblerCheck is used in the example. The example does not have focus on construction of the request and parsing of the response.

The following to example files can be found on the DGA's website.

Documents and files

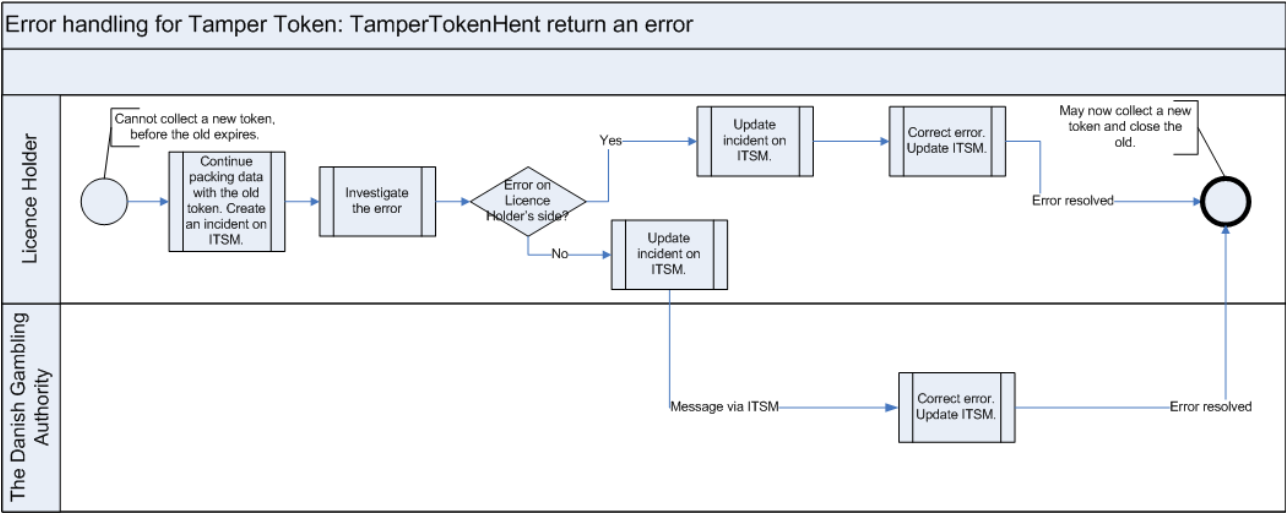
Example in .Net: GamblerServiceExampleClient.cs
 Example in java: GamblerServiceExampleClient.java

4.4. Tamper Token configurations

This section contains descriptions of the error process for Tamper Token and information on the MAC algorithm which the Licence Holder must build.

4.4.1. Error handling for Tamper Token Hent service

Below is an illustration of the proces for handling errors on TamperTokenHent.



Process description

The start condition of the process is the situation, where a Licence Holder cannot collect a new token, before the token, which already is in use, expires.

The Licence Holder should continue to pack data with the old token, even though this token cannot be closed on time. The Licence Holder informs the Danish Gambling Authority through an incident reporting system, if the token cannot be closed before it expires.

The Licence holder investigates whether or not they can correct the error themselves, or if there is an error on the Danish Gambling Authority’s side. The Licence Holder updates the incident via the incident reporting system when it has been identified whether the error is on the Licence Holders side or the Danish Gambling Authority’s side.

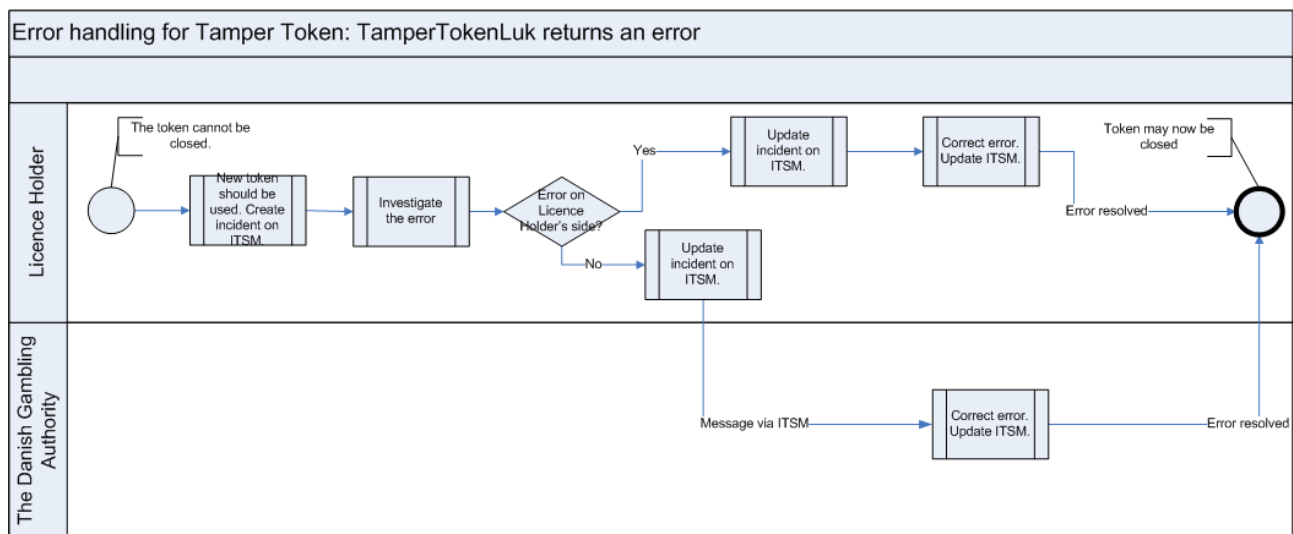
If the error is on the Licence Holder’s side, the Licence Holder should correct the error and update the incident via the incident reporting system.

If the error is on the Danish Gambling Authority's side, the Danish Gambling Authority will correct the error and update the incident via the incident reporting system. The Licence Holder may now see a status via login on the incident reporting system.

The Licence Holder may now collect a new token and close the old hereafter.

4.4.2. Error handling for Tamper Token Luk service

Below is an illustration of the process for handling errors on TamperTokenLuk.



Process description

The start condition of the process is the situation where a Licence Holder cannot close the token on time before it expires.

At this point in the process, the Licence Holder will have collected a new token, since a new token must be collected before closing the existing. The Licence Holder should use the new token during the error handling process for TamperTokenLuk.

The Licence Holder should inform the Danish Gambling Authority via the incident reporting system on the error on TamperTokenLuk.

The Licence Holder investigates whether or not they can correct the error on their side, or whether or not the error is on the Danish Gambling Authority's side. The Licence Holder updates the incident via the incident reporting system.

If the error is on the Licence Holders side, the Licence Holder should correct the error and update the incident via the incident reporting system.

If the error is on the Danish Gambling Authority's side, the Danish Gambling Authority will correct the error and update the incident via the incident reporting system. The Licence Holder may now see an updated status via login on the incident reporting system.

The Licence Holder may now close the token.

4.4.3. Mechanism for MAC generation

The Licence Holder must build a mechanism to generate a MAC in the right way. It must match the way the Gambling Authority has built the same mechanism to ensure that MACs later can be compared. In the following sections it is described how to build this mechanism to generate MAC, including which API to apply.

The process for the mechanism that generates the MAC for a token is illustrated in the figure on page 37.

The 5 points on the figure is described in the following:

- Ad 1. License Holder activates the service operation `TamperTokenHent` and receives a new `TamperTokenID` and a key, `Key`, which must be used for MAC generation for the first Standard Record file for the new Token.
- Ad 2. When a new Standard Record file is generated, a MAC is generated of this record in near real time.
- Ad 3. The generated MAC is now the new `Key` for the MAC generation.
- Ad 4. After the MAC generation the present Standard Record file is added to a zip-file for the present token.
- Ad 5. When a token is closed, typically after the time interval which is given in the response from the service operation `TamperTokenHent`, the service operation `TamperTokenLuk` is activated with ID for the Token which should be closed, the latest generated MAC and the identification of the License Holder.

4.4.4. MAC API

For generation of MACs the class `SecretKeySpec` from Java 1.6 must be applied. Below is presented an example of how the code can look for the step "Generate MAC for Standard Record file using KEY".

The example uses the class `ByteArrayHandler`, which can be found on the Danish Gambling Authority's website.

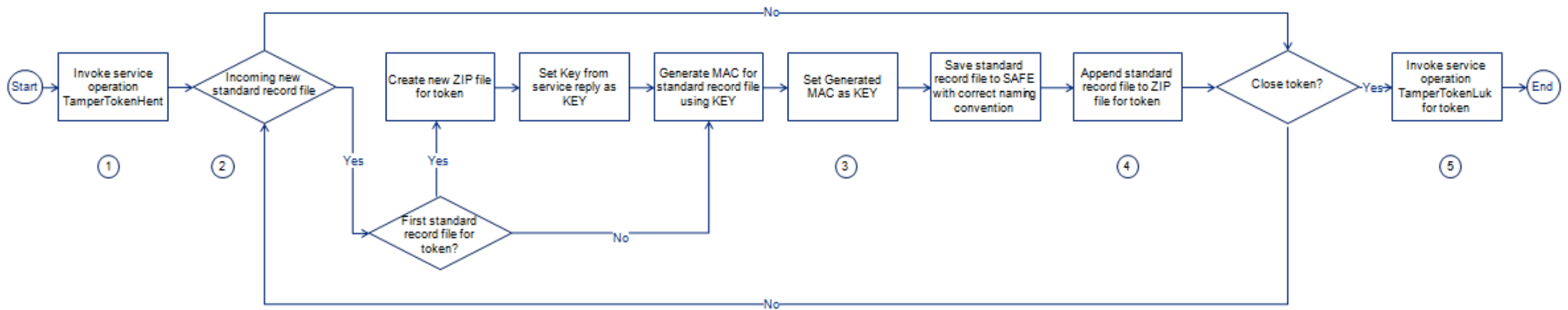
Documents and files
<code>ByteArrayHandler.java</code>

The argument key is either the key from the service operation TamperTokenHent or the MAC from a previous file and InputStream contains the data from a Standard Record, for which to generate a MAC.

Example:

```
public String getMAC(String key, InputStream input) throws TamperTokenException {
    try {
        Mac mac = Mac.getInstance("HmacSHA256");
        byte[] byteKey = ByteArrayHandler.parseString(key);
        SecretKeySpec keySpec = new SecretKeySpec(byteKey, "HmacSHA256");
        mac.init(keySpec);
        byte[] data = new byte[1024];
        int read;
        while((read=input.read(data)) > -1){
            mac.update(data, 0, read);
        }
        return ByteArrayHandler.toString(mac.doFinal());
    }
    catch (Exception e) {
        throw new TamperTokenException(e);
    }
}
```

4.4.5. MAC algorithm – figure



4.4.6. Example of MAC calculation

On the website of the Danish Gambling Authority, there is a link to the folder “Vejledning”:
<http://85.81.229.78/projects/SPIL/Vejledning/>. In this folder is placed a file called TamperTokenTest3-2152.zip, which has been used below to give an example of calculation of MAC.

Documents and files
TamperTokenTest3-2152.zip

The example has been made for SpilCertifikatIdentifikation = TamperTokenTest3 and TamperTokenID = 2152.

The file TamperTokenTest3-2152.zip contains three files: TamperTokenTest3-2152-1.txt, TamperTokenTest3-2152-2.txt and TamperTokenTest3-2152-E.txt. A start MAC is retrieved by the serviceoperation TamperTokenHent, and it is shown below as TamperTokenStartMAC. Following this are shown the intermediate MACs, which is calculated on each file. The MAC from the last file is reported by the serviceoperation TamperTokenLuk in the element TamperTokenMAC.

TamperTokenStartMAC = fb99919c20c57b01a1ab37fdc576f75a

MAC of file TamperTokenTest3-2152-1.txt =
148f1bc4bfe2be67cfed691f6a703ed90e780f45faab665b5c86a3c8346ad056

MAC of file TamperTokenTest3-2152-2.txt =
a79953be54a71069a07d2d7c63566daaab221de984d93c36ae8c7b26d149df90

MAC of file TamperTokenTest3-2152-E.txt =
1b14a1da76568ab3b96bc64bb7ee02e846fbd7711e3ce40f477b0c66a0663016

TamperTokenMAC = 1b14a1da76568ab3b96bc64bb7ee02e846fbd7711e3ce40f477b0c66a0663016

4.4.7. Handling unused tokens

In the case where Licence Holder has used the service TamperTokenHent to retrieve a token, where it later appears that the token should not be used anyway, then the Licence Holder may close this token again by using the serviceoperation TamperTokenLuk and let the field TamperTokenMAC (which normally contains the calculated MAC) contain the string ”empty”. Such a service request will look as follows:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://skat.dk/begrebsmodel/2009/01/15/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:TamperTokenAnvend_ID>
      <ns:Kontekst>
        <ns1:HovedOplysninger xmlns:ns1="http://skat.dk/begrebsmodel/xml/schemas/kontekst/2007/05/31/">
          <ns1:TransaktionsID>895ffb40-9f4a-11e0-8264-0800200c9a66</ns1:TransaktionsID>
          <ns1:TransaktionsTid>2011-10-15T18:41:30.054+01:00</ns1:TransaktionsTid>
        </ns1:HovedOplysninger>
      </ns:Kontekst>
    </ns:TamperOperationValg>
  </soapenv:Body>
</soapenv:Envelope>
```

```

<ns:TamperTokenLuk>
  <ns:TamperTokenID>2112</ns:TamperTokenID>
  <ns:SpilCertifikatIdentifikation>TamperTokenTest3</ns:SpilCertifikatIdentifikation>
  <ns:TamperTokenMAC>empty</ns:TamperTokenMAC>
</ns:TamperTokenLuk>
</ns:TamperOperationValg>
</ns:TamperTokenAnvend_I>
</soapenv:Body>
</soapenv:Envelope>

```

This way of handling unused tokens gives the Licence Holder the possibility for parallel SAFEs, such that the Licence Holder may retrieve two tokens in parallel and use one for the primary SAFE and the other for the secondary SAFE. The secondary token will only be used in the case where an error occurs on the primary SAFE, however, the Licence Holder must still remember to close the token before it expires.

4.5. Examples of Tamper Token and ROFUS service request and response

In the following are presented examples of service requests and responses for the services which the Licence Holder must use. The examples are thought of as help for the Licence Holder to understand the services to be used, but it is not the intention that the Licence Holder should build code based on the examples. For this purpose is referred to the service descriptions, XSD schemas and WSDL-files which are found on the Danish Gambling Authority's website.

4.5.1. TamperTokenHent

Request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://skat.dk/begrebsmodel/2009/01/15/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:TamperTokenAnvend_I>
      <ns:Kontekst>
        <ns1:HovedOplysninger xmlns:ns1="http://skat.dk/begrebsmodel/xml/schemas/kontekst/2007/05/31/">
          <ns1:TransaktionsID>895ffb40-9f4a-11e0-8264-0800200c9a66</ns1:TransaktionsID>
          <ns1:TransaktionsTid>2011-06-25T18:41:30.054+01:00</ns1:TransaktionsTid>
        </ns1:HovedOplysninger>
      </ns:Kontekst>
      <ns:TamperOperationValg>
        <ns:TamperTokenHent>
          <ns:SpilCertifikatIdentifikation>TamperTokenTest3</ns:SpilCertifikatIdentifikation>
        </ns:TamperTokenHent>
      </ns:TamperOperationValg>
    </ns:TamperTokenAnvend_I>
  </soapenv:Body>
</soapenv:Envelope>

```

Response:

```

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <env:Body>
    <ns:TamperTokenAnvend_O xmlns:ns="http://skat.dk/begrebsmodel/2009/01/15/">
      <ns:Kontekst>
        <HovedOplysningerSvar xmlns="http://skat.dk/begrebsmodel/xml/schemas/kontekst/2007/05/31/">
          <TransaktionsID>895ffb40-9f4a-11e0-8264-0800200c9a66</TransaktionsID>
          <ServiceID>TamperTokenAnvendService</ServiceID>
          <TransaktionsTid>2011-06-25T18:41:30.054+01:00</TransaktionsTid>
        </HovedOplysningerSvar>
      </ns:Kontekst>
    </ns:TamperTokenAnvend_O>
  </env:Body>
</env:Envelope>

```

```

    </HovedOplysningerSvar>
  </ns:Kontekst>
</ns:TamperTokenHent_O>
  <ns:TamperTokenID>385</ns:TamperTokenID>
  <ns:TamperTokenStartMAC>a06174fd062bb397894860bd5c20aa08</ns:TamperTokenStartMAC>
  <ns:TamperTokenUdstedelseDatoTid>2011-06-25T18:47:04.481+02:00</ns:TamperTokenUdstedelseDatoTid>
  <ns:TamperTokenPlanlagtLukketDatoTid>2011-06-26T18:47:04.481+02:00</ns:TamperTokenPlanlagtLukketDatoTid>
</ns:TamperTokenHent_O>
</ns:TamperTokenAnvend_O>
</env:Body>
</env:Envelope>

```

4.5.2. TamperTokenLuk

Request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://skat.dk/begrebsmodel/2009/01/15/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:TamperTokenAnvend_I>
      <ns:Kontekst>
        <ns1:HovedOplysninger xmlns:ns1="http://skat.dk/begrebsmodel/xml/schemas/kontekst/2007/05/31/">
          <ns1:TransaktionsID>895ffb40-9f4a-11e0-8264-0800200c9a66</ns1:TransaktionsID>
          <ns1:TransaktionsTid>2011-06-25T18:41:30.054+01:00</ns1:TransaktionsTid>
        </ns1:HovedOplysninger>
      </ns:Kontekst>
      <ns:TamperOperationValg>
        <ns:TamperTokenLuk>
          <ns:TamperTokenID>385</ns:TamperTokenID>
          <ns:SpilCertifikatIdentifikation>TamperTokenTest3</ns:SpilCertifikatIdentifikation>
          <ns:TamperTokenMAC>2da9fe732840bc40f05eefbace7bf03fc36e141907a8d6ce7da329fa0f1bb25c
          </ns:TamperTokenMAC>
        </ns:TamperTokenLuk>
      </ns:TamperOperationValg>
    </ns:TamperTokenAnvend_I>
  </soapenv:Body>
</soapenv:Envelope>

```

Response:

```

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <env:Body>
    <ns:TamperTokenAnvend_O xmlns:ns="http://skat.dk/begrebsmodel/2009/01/15/">
      <ns:Kontekst>
        <HovedOplysningerSvar xmlns="http://skat.dk/begrebsmodel/xml/schemas/kontekst/2007/05/31/">
          <TransaktionsID>895ffb40-9f4a-11e0-8264-0800200c9a66</TransaktionsID>
          <ServiceID>TamperTokenAnvendService</ServiceID>
          <TransaktionsTid>2011-06-25T18:41:30.054+01:00</TransaktionsTid>
          <SvarReaktion>
            <Advis>
              <AdvisNummer>0</AdvisNummer>
              <AdvisTekst>Token is now closed</AdvisTekst>
              <ServiceID>TamperTokenAnvendService</ServiceID>
            </Advis>
          </SvarReaktion>
        </HovedOplysningerSvar>
      </ns:Kontekst>
    </ns:TamperTokenAnvend_O>
  </env:Body>
</env:Envelope>

```


4.5.3. GamblerCheck for person nummer not registered in ROFUS

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://services.lur.skat.dk">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GamblerCheckRequest>
      <Kontekst>
        <ns1:HovedOplysninger xmlns:ns1="http://skat.dk/begrebsmodel/xml/schemas/kontekst/2007/05/31/">
          <ns1:TransaktionsID>f0e33d70-9f4c-11e0-8264-0800200c9a66</ns1:TransaktionsID>
          <ns1:TransaktionsTid>2011-06-25T19:33:41.002+01:00</ns1:TransaktionsTid>
        </ns1:HovedOplysninger>
      </Kontekst>
      <PersonInformation>
        <PersonCPRNumber>1110911013</PersonCPRNumber>
      </PersonInformation>
    </ser:GamblerCheckRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:GamblerCheckResponse xmlns:ns3="http://services.lur.skat.dk"
  xmlns:ns2="http://skat.dk/begrebsmodel/xml/schemas/kontekst/2007/05/31/">
      <Kontekst>
        <ns2:HovedOplysningerSvar xmlns="http://skat.dk/begrebsmodel/xml/schemas/kontekst/2007/05/31/">
          <TransaktionsID>f0e33d70-9f4c-11e0-8264-0800200c9a66</TransaktionsID>
          <ServiceID>LUR_SPILLER_SOEG</ServiceID>
          <TransaktionsTid>2011-06-25T19:34:20.020+02:00</TransaktionsTid>
        </ns2:HovedOplysningerSvar>
      </Kontekst>
      <ExclusionStatus>
        <GamblerExclusionStatus>PersonIsNotRegistered</GamblerExclusionStatus>
      </ExclusionStatus>
    </ns3:GamblerCheckResponse>
  </S:Body>
</S:Envelope>
```

4.5.4. GamblerCheck for person nummer temporarily registered in ROFUS

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://services.lur.skat.dk">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GamblerCheckRequest>
      <Kontekst>
        <ns1:HovedOplysninger xmlns:ns1="http://skat.dk/begrebsmodel/xml/schemas/kontekst/2007/05/31/">
          <ns1:TransaktionsID>f0e33d70-9f4c-11e0-8264-0800200c9a66</ns1:TransaktionsID>
          <ns1:TransaktionsTid>2011-06-25T19:40:41.002+01:00</ns1:TransaktionsTid>
        </ns1:HovedOplysninger>
      </Kontekst>
      <PersonInformation>
        <PersonCPRNumber>1110913016</PersonCPRNumber>
      </PersonInformation>
    </ser:GamblerCheckRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
```

```

<S:Body>
  <ns3:GamblerCheckResponse xmlns:ns3="http://services.lur.skat.dk"
xmlns:ns2="http://skat.dk/begrebsmodel/xml/schemas/kontekst/2007/05/31/">
    <Kontekst>
      <ns2:HovedOplysningerSvar xmlns="http://skat.dk/begrebsmodel/xml/schemas/kontekst/2007/05/31/">
        <TransaktionsID>f0e33d70-9f4c-11e0-8264-0800200c9a66</TransaktionsID>
        <ServiceID>LUR_SPILLER_SOEG</ServiceID>
        <TransaktionsTid>2011-06-25T19:41:50.751+02:00</TransaktionsTid>
      </ns2:HovedOplysningerSvar>
    </Kontekst>
    <ExclusionStatus>
      <GamblerExclusionStatus>PersonIsRegisteredTemporarily</GamblerExclusionStatus>
    </ExclusionStatus>
  </ns3:GamblerCheckResponse>
</S:Body>
</S:Envelope>

```

4.5.5. GamblerCheck for person nummer indefinitely registered in ROFUS

Request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://services.lur.skat.dk">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GamblerCheckRequest>
      <Kontekst>
        <ns1:HovedOplysninger xmlns:ns1="http://skat.dk/begrebsmodel/xml/schemas/kontekst/2007/05/31/">
          <ns1:TransaktionsID>f0e33d70-9f4c-11e0-8264-0800200c9a66</ns1:TransaktionsID>
          <ns1:TransaktionsTid>2011-06-25T19:44:41.002+01:00</ns1:TransaktionsTid>
        </ns1:HovedOplysninger>
      </Kontekst>
      <PersonInformation>
        <PersonCPRNumber>1110910017</PersonCPRNumber>
      </PersonInformation>
    </ser:GamblerCheckRequest>
  </soapenv:Body>
</soapenv:Envelope>

```

Response:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:GamblerCheckResponse xmlns:ns3="http://services.lur.skat.dk"
xmlns:ns2="http://skat.dk/begrebsmodel/xml/schemas/kontekst/2007/05/31/">
      <Kontekst>
        <ns2:HovedOplysningerSvar xmlns="http://skat.dk/begrebsmodel/xml/schemas/kontekst/2007/05/31/">
          <TransaktionsID>f0e33d70-9f4c-11e0-8264-0800200c9a66</TransaktionsID>
          <ServiceID>LUR_SPILLER_SOEG</ServiceID>
          <TransaktionsTid>2011-06-25T19:45:14.800+02:00</TransaktionsTid>
        </ns2:HovedOplysningerSvar>
      </Kontekst>
      <ExclusionStatus>
        <GamblerExclusionStatus>PersonIsRegisteredIndefinitely</GamblerExclusionStatus>
      </ExclusionStatus>
    </ns3:GamblerCheckResponse>
  </S:Body>
</S:Envelope>

```

5. Licence Holder's quality assurance

It is the responsibility of the Licence Holder to ensure compliance with all technical requirements. This section describes three areas which Licence Holder must be particularly aware of. In section 5.1 is described a check-list, which Licence Holder must complete in order to transfer from version 1 to version 2 of Standard Records. In section 5.2 is described how Licence Holder may ensure that the generated Standard Records are schema valid and in section 5.3 is described how Licence Holder should ensure the Tamper Token MAC is calculated correctly.

5.1. Checklist for new version of Standard Records

The DGA has produced a checklist including all changes between version 1 and 2 of Standard Records. The Licence Holder must complete this checklist when transferring from version 1 to 2 and send this to the Danish Gambling Authority. The completed checklist must be e-mailed to tekniske.spm@skat.dk before reporting the first Standard Record on the new version.

Documents and files

Ændringer til Standard Records – 12mar2012.xls
Ændringer til Standard Records – english – 12mar2012.xls

5.2. Schema validation of Standard Records using XML Spy

A lot of data errors can be caught by ensuring that the reported data are schema valid. The schema validation must be checked by Licence Holder, and it can be done using the programme XML Spy in the following way:

1. Download and install XML Spy
 - a. Download a fully functional free 30-day trial-version of Altova XMLSpy 2012 (http://www.altova.com/download/xmlspy/xml_editor_enterprise.html)
2. Download XML skeme files from the Danish Gambling Authority's website
 - a. See section 2.2 for reference to file name.
 - b. Unzip
3. Start a new project in XML Spy
 - a. Use explorer to drag your unzipped XML skeme files into the "DTD/Schemas" folder in the Project navigation worksheet.
4. You are now ready to validate a file,
 - a. Open the XML file in XML Spy
 - b. Assign the schema to the file
 - i. Choose the DTD/Schema menu, option 2 – Assign Schema
 - ii. Choose ok
 - iii. Choose window
 - iv. Choose the correct XSD file from the list
 - v. Choose ok, ok, ok (3 times)

- vi. Hit F8 to validate your XML fil, the result is presented in the Messages window.
5. Correct any errors and make sure that all xml-files are valid before they are uploaded to SAFE.

5.3. Validation of correct Tamper Token MAC calculation

It is important that the Tamper Token MAC code is calculated correctly. Licence Holder must ensure that if the MAC code of a zip-file which the Licence Holder has placed on SAFE is calculated, the value must match the value which Licence Holder has reported to the Tamper Token system.

Licence Holder must perform a thorough test and ensures that the reported MAC also corresponds to the MAC which can be calculated from the file.