Prepared by	Document id	Version	Document date
WM-data	RKF-0024	1.0	31 May 2001

RKF Travel Card
Implementation Guide Type 1

CONTENTS

1 INTRODUCTION

1.1 Scope

This document contains guidelines for implementing the type 1 RKF travel card as specified in [RKF-0022] and [RKF-0023], i.e. a contactless memory card in accordance with [ISO 14443], type A of size 1 kB.

The purpose of this guide is to give advice on design of travel cards and travel card systems in accordance with the specification.

Every implementation of the type 1 RKF travel card should, comprise a description of deviations from the guidelines of this document.

1.2 Reader's Guide

It is recommended, that this document is read "in parallel" with the implementation specification in [RKF-0022] and [RKF-0023].

A knowledge of the contents of the requirement specification [RKF-0020] is required.

In this document, names of data elements and data element groups are written in italics using the Arial font, e.g.: *TCPU: Purse.*

2 GENERAL TRANSACTION PROCEDURE

This chapter contains a short introduction to the way a general transaction procedure for normal validation transactions in a front system could be constructed.

The scope of this chapter is not to give a detailed specification of how to design and construct this part of a front system. The intention is rather to provide an understanding of how the elementary actions of a complete transaction can be assembled in order to achieve the necessary functionality and to achieve reasonable transaction times.

It is assumed, that the type of transaction (e.g. 'check-in', 'check-in with specified destination', 'purse update') is known prior to performing the procedure.

2.1 Overview

The transaction procedure is performed in these steps:

A. Read basic card information

B. Determine TCAS

C. Determine relevant application objects and fare calculation

D. Read relevant applications

E. Perform fare calculation

F. Write changed sectors/blocks

2.2 Procedure

A. Read basic card information

- Sectors 0 and 1 are read from the card: CMI, TCCI, TCAS (1), TCDI.
- The *CardSerialNumber* is checked against a possible list of blocked cards.
- The directory TCDI is scanned in order to find the position of TCAS (2)
- TCAS (2) is read from the card
- Relevant checksum and MAC values are recalculated and controlled.

B. Determine TCAS

- TCAS (1) and TCAS (2) are checked for consistency, and the valid instance is determined according to [RKF-0022] (section 'TCAS: Read Function').

C. Determine relevant application objects and fare calculation

- The directory is scanned, and for each sector it is checked, if the combination of AID, PIX identifies an application object, that is valid for the transaction type.
- If this process does not result in a match:

  There is an error and the card can not be processed.
- If the process results in a match for just one sector:

  This application object becomes the primary application object in this transaction.
- If the process results in a match for more than one sector:

  The primary application object in this transaction is determined by a priority mechanism.
- For the primary application object, mandatory and optional supplementary application objects are determined. It is an error, if mandatory application objects are not present on the card.
- For the primary application object, the method of fare calculation is determined.

D. Read relevant applications

- The sectors/blocks of the relevant application objects determined in step C are read from the card. For each application object that contains dynamic parts, the relevant blocks are located based on the information of TCAS.
- Relevant checksum and MAC values are recalculated and controlled.

E. Perform fare calculation

- The method of fare calculation determined in step C is performed based on the application objects read in step D. Changed sectors/blocks are not written to the card in this step.

F. Write changed sectors/blocks

- Changed sectors/blocks are written to the card after calculating relevant checksums and MAC values.
- TCAS is updated to reflect the changes of application objects and the two instances are written in this order: TCAS (1), TCAS (2).
- All block writes are followed by control readings.

3 BACKWARDS COMPATIBILITY

When the travel card is used for a long period of time, a need for changes of the travel card structure and contents will naturally arise. New possibilities and demands must be dealt with.

When carrying out this kind of changes, it is important to achieve a high level of backward compatibility in order to minimise the disturbance of the passengers and to minimise the required resources.

3.1 General

Changes to the travel card structure and/or contents can roughly be divided into 3 types of changes:

1. New version of an application object
2. Structural change
3. New card technology

This RKF travel card specification contains tools to manage changes:

- The *TCCI: Card Information* of the travel card support layer has a data element *CardVersion*, that is a superior version number to all system and application objects.
- Every application object, and most system objects contain a *VersionNumber* data element.

The superior *CardVersion* number is increased, if fundamental changes to the travel card structure are carried out. E.g. changes to *TCDI: Directory* or *TCAS: ApplicationsStatus*.

The *VersionNumber* of a system or application object is increased, if application objects are changed. E.g. data elements are added, changed or removed, possibly resulting in a change of size of the system or application object.

Both types of version numbers have a valid range 0..63.

Resekortföreningen controls the assignment of new version numbers.

The remainder of this section describes how changes can be handled.

3.1.1 New Version of an Application Object

This type of change comprises a new version of an application object, while the overall structure of the card remains unchanged. This could for example be a change in the size of a data element or the introduction of a new data element.

Sometimes the change will only influence one application object. In other cases the changes will include a co-ordinated change of several application objects. For example a change in the size of data elements containing money amounts would lead to new versions of both the TCPU, TCTI and the TCCO.

A change can be based upon a need within one or more PTAs. For each change it is important to consider whether the existing interoperability agreements can be used in their present form, or whether they have to be changed. Unless this problem is considered carefully, any interoperability that have been achieved could easily be lost.

The process of carrying through a changing could contain the following steps :

1. The change of the application object is designed and described.
2. The change is approved by Resekortsföreningen and the other PTAs using the application objects in compliance with the interoperability agreements.
3. Front system software is altered so that it can process travel card with both the old and the new version of the application object.
4. The new software for the front systems is distributed over a period of time together with a date and time specifying when to activate the new software.
5. When the date and time distributed with the new software occurs, the new version of the software is activated. After that the front systems will alter the cards to update to the new version of the application object when handling a card.
6. A later update of the front system software can remove the possibility of using the old version of the application object.

For some types of changes it will not be desirable to perform an automatic change to a new version of the application object. In this case the altered software must be able to handle both the old and new version.

3.1.2 Structural Change

This type of change concerns the basic relations between several application objects (e.g. the relation between *TCEL: Event Log* and TCTI/TCCO).

In some cases a change can be performed by introducing new versions of the relevant application objects combined with a change process similar to the one described in section 3.1.1.

In other cases the changes are so radical, that a new card version number must be used in the *TCCI: Card Information*. In each case it must be analysed and decided how backwards compatibility can be ensured. Often the proces of change can be performed like in section 3.1.1.

In the case of very large changes, it can be necessary that all cards in circulation are upgraded either manually or in specially designed machines.

3.1.3 New Card Technology

This type of change concerns the introducing of a new card technology. It is not possible to determine the degree of backwards compatibility that can be obtained, before the actual technology has been determined.

When a new card technology is introduced, this specification must be expanded to handle the issues of the new technology.

I some cases, a new card technology will utilise the same form of communication as the present card technology. In these cases, it will be possible to carry out the changes over a period of time with very little or no bother for the passengers.

In other cases, the new technology will imply a new form of communication. However, if the front system CADs are prepared for this new form of communication, it will still be possible to make the changes over a period of time while bothering the passengers as little as possible.

3.2 From Travel Card Specification Volume A to Volume B

The change from volume A of this specification to the present volume B is a 'structural change' according to the types of changes described in section 3.1. The major changes are:

- *TCCI: Card Information* is changed. *CardVersion* is increased from 0 to 1.
- *TCEL: Event Log* is changed.
- *TCAS: Applications Status* is changed.
- The use of *TCAS: Applications Status* as the key element for ensuring indivisibility of transactions. TCAS is duplicated in two instances.

When upgrading from volume A to volume B, it is therefore not possible to upgrade each travel card "on-the-fly" as a part of normal validation transactions, as all system and application objects of the travel card must be reinitiated. The physical travel card can be reused, but it will be necessary to perform the upgrade at special front systems either by self-service or by service personnel.

By appropriate design of the front system software, it will be possible to perform the upgrading of all travel cards over a period of time of desired length.

4 INTEROPERABILITY

If the same travel cards are to be used by customers across two or more PTAs, it is necessary to ensure interoperability of the travel card between these PTAs. I.e. it must be well-defined to what extent travel cards and system objects and application objects issued by one PTA can be used by other PTAs.

This travel card specification does not in itself guarantee interoperability. A group of co-operating PTAs must agree on the desired level of interoperability and take the necessary measures to achieve that level.

Section 4.1 describes 5 levels of interoperability intended to facilitate the definition of interoperability.

Section 4.2 outlines how an application object created by one PTA can be removed by other PTAs.

Section 4.3 shortly describes how agreements on interoperability could be established.

The backwards compatibility issue described in chapter 3 is closely connected to the issue of interoperability.

4.1 Levels of Interoperability

5 levels of interoperability are defined. The higher the level, the higher extent of interoperability.

Short description of the interoperability levels:

Interoperability level	Short description
1: Common card technology	- The same card technology is used. PTAs can include application objects on each others cards, but a PTA cannot use an application object of another PTA.
2. Common implementation framework	- The card technology is used in accordance with the relevant *Implementation Specification* (e.g. [RKF-0022] for type 1 cards). - It is possible to define common application objects, that are handled properly as regards transactions and security. - Level 1 interoperability is assumed.
3: Common TCPU	- The electronic purse TCPU is shared by a group of PTAs. - If level 5 interoperability is not agreed upon, the TCPU can only be used for simple payment of services in other PTAs. - Level 2 interoperability is assumed.

4: Accept tickets/contracts	- A PTA is able to read and interpret the contents of application objects (TCTI, TCCO etc.) of other PTAs. The utilisation of the contents can vary:
	  - Show data
	  - Simple control validation
	  - Data is used as input to fare calculation
	- Level 2 interoperability is assumed.

5: Common tickets/contracts	- A PTA is able to update application objects (TCTI, TCCO etc.) of other PTAs. - The application objects of different PTAs can work together in a truly integrated manner. - Level 3 and 4 interoperability is assumed.

The relation between interoperability levels is outlined in the following figure 1 (an arrow from level x to level y meaning level y a prerequisite to level x):

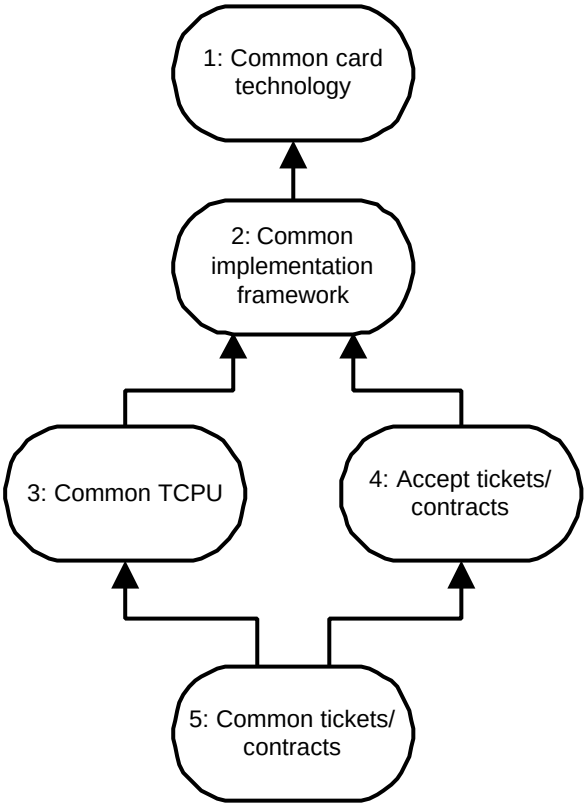

Figure 1: Interoperability levels relations

Detailed description of the interoperability levels:

Interoperability level	Detailed description
1: Common card technology	Prerequisites: - Physical card based on the same card technology and communication protocol To be agreed upon: - Handling of common access keys for relevant sectors

2. Common implementation framework	Prerequisites:
	- Level 1 interoperability
	- The travel cards must be used in accordance with the relevant *Implementation Specification* (e.g. [RKF-0022] for type 1 cards) as regards:
	  - Contents of and rules for system objects and application objects (cf. section 'General Rules for Travel Card Data' of [RKF-0022])
	  - Coding of data elements (cf. section 'Rules for Coding Data Elements' of [RKF-0022])
	  - Checksum and MAC handling (cf. section 'Calculation of Checksum and MAC Values' of [RKF-0022])
	  - Ensuring indivisible transactions using the TCAS (cf. section 'Transaction Concept' of [RKF-0022])
	To be agreed upon:
	- Allowed versions of relevant application objects
	- Handling of common MAC keys

3: Common TCPU	Prerequisites: - Level 2 interoperability - The travel cards must be used in accordance with the relevant *Implementation Specification* (e.g. [RKF-0022] for type 1 cards) as regards: - Contents of and rules for TCPU To be agreed upon: - Allowed versions of TCPU - Handling of TCPU
4: Accept tickets/contracts	Prerequisites: - Level 2 interoperability - The travel cards must be used in accordance with the relevant *Implementation Specification* (e.g. [RKF-0022] for type 1 cards) as regards: - Contents of and rules for relevant application objects (TCEL, TCTI, TCCO, TCST, TCCP, TCDB) To be agreed upon: - Allowed versions of relevant application objects - Interpretation of relevant application objects

Interoperability level	Detailed description
5: Common tickets/contracts	Prerequisites: - Level 3 and 4 interoperability To be agreed upon: - Handling of relevant application objects

4.2 Removal of Applications Created by Other PTAs

If a customer uses the travel card in several areas belonging to different PTAs, it will typically be necessary to create several ticket and/or contract application objects on the travel card. Sooner or later, all sectors of the travel card will be occupied, and the need to remove (or move) one or more application objects will arise.

If the application object to remove is provided by the PTA performing the update of the application objects of the card, this will be easily done, as the relevant access keys and MAC keys are known, as well as the conditions that must be satisfied before removing the object, e.g. repurchase of any values left or check expiration date.

If, on the other hand, the application object is provided by another PTA than the PTA performing the update of the application objects of the card, the information needed for removing the object might not be available.

This problem can be solved in a number of ways:

- The information needed for removing application objects is distributed to other PTAs in a secure way.
- It is made possible for a PTA to manipulate a travel card remotely using a CAD located at another PTA. E.g. a Stockholm PTA could remove a ticket from a card presented to a CAD in Copenhagen.
- Small software components, that can perform the removal of application objects, are distributed in a secure way among relevant PTAs. The software comprises access keys, MAC keys, and relevant conditions that must be satisfied in order to perform the removal properly.
- If the removal problem only happens very infrequently, it might be solved by repurchasing any values left on the travel card succeeded by the issuing of a new card.

The handling of the removal problem must be a part of relevant interoperability agreements.

4.3 Interoperability Agreements

If a group of co-operating PTAs wish to obtain one of the interoperability levels described in section 4.1, an interoperability agreement must be established.

The process of establishing an interoperability agreement should profit by utilising the levels of interoperability defined in section 4.1, and the considerations in section 4.2.

An interoperability agreement could be a kind of matrix, that for all relevant pairs of PTAs specifies the interoperability level and other relevant conditions.

Interoperability agreements should at a suitable level of details be available to all PTAs co-operating in the RKF, and might create changes and/or amendments to this specification.

5 TYPICAL CARD SAMPLES

This chapter outlines two typical examples of travel cards.

5.1 Card Sample with TCEL, TCPU, TCTI, TCCO

This card contains the following system and application objects:

System/application object	Comments
CMI: Manufacturer Information	Mandatory
TCCI: Card Information	Mandatory
TCAS: Applications Status (1)	Mandatory
TCDI: Directory	Mandatory
TCAS: Applications Status (2)	Mandatory, positioned in separate sector
TCEL: Event Log	9 log records
TCPU: Purse	
TCTI: Ticket	Static data: 2 blocks, dynamic data: 3 blocks
TCCO: Contract	Static data: 3 blocks, dynamic data: 1 block

The following figure outlines a possible layout of the travel card:

	B0	B1	B2
S0	CMI	TCCI	TCAS (1)
S1	TCDI (1/3)	TCDI (2/3)	TCDI (3/3)
S2	TCAS (2)		
S3	TCEL Record (1)	TCEL Record (2)	TCEL Record (3)
S4	TCEL Record (4)	TCEL Record (5)	TCEL Record (6)
S5	TCEL Record (7)	TCEL Record (8)	TCEL Record (9)
S6	TCPU, Static Data	TCPU, Dyn. Data (1)	TCPU, Dyn. Data (2)
S7	TCTI, Static Data (1/2)	TCTI, Static Data (2/2)	TCTI, Dyn. Data (1,1/3)
S8	TCTI, Dyn. Data (1,2/3)	TCTI, Dyn. Data (1,3/3)	TCTI, Dyn. Data (2,1/3)
S9	TCTI, Dyn. Data (2,2/3)	TCTI, Dyn. Data (2,3/3)	
S10	TCCO, Static Data (1/3)	TCCO, Static Data (2/3)	TCCO, Static Data (3/3)
S11	TCCO, Dyn. Data (1)	TCCO, Dyn. Data (2)	
S12			
S13			
S14			
S15			

5.2 Card Sample with TCEL, TCPU, TCST-ci/co, TCDB, TCCP

This card contains the following system and application objects:

System/application object	Comments
CMI: Manufacturer Information	Mandatory
TCCI: Card Information	Mandatory
TCAS: Applications Status (1)	Mandatory
TCDI: Directory	Mandatory
TCEL: Event Log	6 log records
TCAS: Applications Status (2)	TCAS (2)
TCPU: Purse	
TCST-ci/co: Special ticket	6 tickets in ticket/log area
TCDB: Discount Basis	
TCCP: Customer Profile	

The following figure outlines a possible layout of the travel card:

	B0	B1	B2
S0	CMI	TCCI	TCAS (1)
S1	TCDI (1/3)	TCDI (2/3)	TCDI (3/3)
S2	TCEL Record (1)	TCEL Record (2)	TCEL Record (3)
S3	TCEL Record (4)	TCEL Record (5)	TCEL Record (6)
S4	TCAS (2)		
S5	TCPU, Static Data	TCPU, Dyn. Data (1)	TCPU, Dyn. Data (2)
S6	TCST-ci/co (1, 1/3)	TCST-ci/co (1, 2/3)	TCST-ci/co (1, 3/3)
S7	TCST-ci/co (2, 1/3)	TCST-ci/co (2, 2/3)	TCST-ci/co (2, 3/3)
S8	TCST-ci/co (3, 1/3)	TCST-ci/co (3, 2/3)	TCST-ci/co (3, 3/3)
S9	TCST-ci/co (4, 1/3)	TCST-ci/co (4, 2/3)	TCST-ci/co (4, 3/3)
S10	TCST-ci/co (5, 1/3)	TCST-ci/co (5, 2/3)	TCST-ci/co (5, 3/3)
S11	TCST-ci/co (6, 1/3)	TCST-ci/co (6, 2/3)	TCST-ci/co (6, 3/3)
S12	TCDB, Static Data	TCDB, Dyn. Data (1)	TCDB, Dyn. Data (2)
S13	TCCP (1/2)	TCCP (2/2)	
S14			
S15			

6 TRAVEL CARD DESIGN GUIDELINES

This chapter is intended as a checklist of issues that should be considered when designing the travel card to be used by a PTA or by a number of co-operating PTAs. It is not intended as a travel card design "cookbook".

Remember, that requirements to the implementation are defined in [RKF-0022] and [RKF-0023].

The guidelines are structured in a set of general guidelines and a set of guidelines for each layer of the RKF travel card.

6.1 General Guidelines

AID values

The need for new AID values to identify PTAs or relevant groups of PTAs must be identified. These AID values will complement the values defined in [RKF-0022] and [RKF-0019]

AID values must be issued by Resekortsföreningen.

PIX values

The need for new PIX values to identify types of tickets, contracts etc. within relevant AIDs must be identified.

Each PTA or group of PTAs (i.e. each AID value) can decide which PIX values to use, and what they represent. There are, however, a few reserved values (see [RKF-0022]).

Positioning of system and application objects

Since the directory is intended to be read in its full length each time a transaction is made on the card, and since system and application objects occupy consecutive sectors, system and application objects can be positioned anywhere in sector 3 to 15.

There are, however, restrictions on the use of sectors 0 to 2.

MAC

The usage of MAC key identifiers to identify valid MAC keys must be determined. This must be done in co-operation with PTAs with whom an interoperability agreement is entered.

If the need for an alternative MAC algorithm (other than desMAC) is identified, this must be decided in co-operation with Resekortsföreningen.

PTA specific data types and data elements

The interpretation of PTA specific data types and data elements must be described in detail.

Interoperability

In order to achieve interoperability with partners with whom co-operation is desired, it is necessary to enter an interoperability agreement to determine relevant application objects and the detailed interpretation of data elements etc.

Reasonable transaction times

In order to ensure reasonable transaction times for normal validation transactions, it is important to pay attention to the co-ordination between the structure and contents of the card and the algorithms used for reading and updating information on the travel card. It is imperative that the number of block reads and block writes must be minimised. The impact of different designs can be calculated by using the calculation models in chapter 7.

New objects or new versions of objects

New versions of or new types of system or application objects must be approved by Resekortsföreningen.

It is important to ensure, that the changes are in accordance with the implementation rules of [RKF-0022]. It is particularly important, that the TCAS is able to ensure indivisible transactions.

6.2 Card Issuer Layer Guidelines

No guidelines available.

6.3 Travel Card Support Layer Guidelines

TCCI: Travel Card Card Information

Currency unit should be selected carefully taking into account in which countries the travel card would possibly be used.

It should also be carefully considered what fraction of the chosen currency is to be used for representing amounts. If, for example, the card is inteded for use in both Sweden and Denmark, it would be appropriate to use 1/100 of SEK (öre) or 1/100 of DKK (øre) as the choice of unit.

TCAS: Travel Card Application Status

It must be decided, which sector the second instance of the TCAS is positioned in. The TCAS(2) will be put in the last block of the chosen sector.

TCEL: Travel Card Event Log

It must be decided, if it is relevant to include an event log, and if so, how many log records are necessary. This decision depends on division of responsibility between the travel card, the front systems, and the back office system.

It is must further be identified, if new event codes are necessary. New event code values must be approved by Resekortsföreningen.

6.4 Travel Card Applications Layer Guidelines

TCPU: Travel Card Purse

Within the purse it is possible to use *DataPointer* as a reference to a PTA specific data area. It must be decided, if this possibility will be used. This possibility is connected to the interoperability issue and must be considered carefully.

TCTI/TCCO: Travel Card Ticket/Contract

The ticket/contract application objects contain a wide selection of possibilities. The static and dynamic parts can be made up of a collection small predefined data element groups. The final design must very carefully reflect the use situation, for which it is designed.

TCST-ci/co: Travel Card Special Ticket ci/co

The size of the ticket/log area, and thereby the number of log entries that can be stored, must be decided.

TCDB: Travel Card Discount Basis

It is necessary to determine the interpretation of each of the three sets of counters and of each of the three time periods that can be stored in the discount basis.

TCST-x: Travel Card Special Ticket

It must be decided, if one or more new special ticket application objects are needed. New CPTS-x application objects must be approved by Resekortsföreningen.

7 TRANSACTION TIMES CALCULATIONS MODELS

This chapter describes calculation models, that can be used to predict obtainable transaction times.

Reliable calculation models are very useful in the process of designing the travel card to be used by a PTA or by a number of co-operating PTAs. The calculation models will help balance the often contradicting demands for acceptable transaction times and for sufficient travel card data.

Vendors of travel card equipment should establish calculation models that apply to their specific equipment. These specific calculation models should be based on the models described in this chapter.

7.1 General Calculation Model

Travel card transaction times are defined:

Total transaction time (TT_{Tot}):

> Denotes the total time from the customer enters the travel card into the read/write field of the card accepting device (CAD) until the transaction is completed, and she can remove the card from read/write field.

I/O transaction time ($TT_{I/O}$):

> Denotes the time used to send data to the travel card and the time to receive data from the travel card. All communication overhead like 'identification phase', 'anti collision', and 'authentication' is included . The necessary processor time of the front system is included. Processor time to calculate checksum and MAC values is not included.

Processor transaction time (TT_{CPU}):

> Denotes the time used for data processing. Includes time to interpret the travel card system objects, time to determine relevant application objects, time to check list of cards to be blocked, and time for fare calculation, checksum calculation, and MAC calculation.

The formula of the general calculation model states, that the total transaction time is the sum of the I/O transaction time and the processor transaction time:

$$TT_{Tot} = TT_{I/O} + TT_{CPU}$$

7.2 I/O Transaction Time Model

The I/O transaction time model is based on the following characteristics of a travel card transaction:

- B_R: Number of read blocks
- S_R: Number of sectors, where one or more blocks are read
- B_W: Number of written blocks
- S_W: Number of sectors, where one or more blocks are written

The formula of the I/O transaction time model further includes constants c_0, c_1, c_2, c_3 and c_4:

$$TT_{I/O} = c_0 + B_R \times c_1 + S_R \times c_2 + B_W \times c_3 + S_W \times c_4$$

Theoretical considerations suggest that the constants of the formula should be (time measured in ms):

$$TT_{I/O} = 3{,}0 + B_R \times 2{,}5 + S_R \times 4{,}0 + B_W \times 8{,}5 + S_W \times 8{,}0$$

Empirical measurements conducted by Resekortsföreningen suggest that the constants of the formula should be (time measured in ms):

$$TT_{I/O} = B_R \times 3{,}0 + S_R \times 5{,}0 + B_W \times 8{,}6 + S_W \times 5{,}0$$

Both formulas assume that every writing of a block is followed by a control reading (cf. section 2.2).

7.3 Processor Transaction Time Model

No formula for the processor transaction time model is supplied in this guide, as this will depend heavily on the characteristics of equipment supplied by vendors. E.g.:

- Design of the front system hardware
- Processor type of the front system
- Design of the front system software
- The degree of optimisation of the front system software
- Complexity of the fare calculation algorithms