FIGI

FINANCIAL INCLUSION
GLOBAL INITIATIVE

SECURITY, INFRASTRUCTURE AND TRUST WORKING GROUP

Methodology for inter-operator and cross-border P2P money transfers

REPORT OF THE QUALITY OF SERVICE WORKSTREAM

Security, Infrastructure and Trust Working Group

Methodology for inter-operator and cross-border P2P money transfers

DISCLAIMER

The Financial Inclusion Global Initiative (FIGI) is a three-year program implemented in partnership by the World Bank Group (WBG), the Committee on Payments and Market Infrastructures (CPMI), and the International Telecommunication Union (ITU), funded by the Bill & Melinda Gates Foundation (BMGF) to facilitate the implementation of country-led reforms to attain national financial inclusion targets, and ultimately the global 'Universal Financial Access 2020' goal. FIGI funds initiatives in three countries-China, Egypt and Mexico; supports working groups to address three distinct challenges for reaching universal financial access:

(1) the Electronic Payment Acceptance Working Group (led by the WBG),
(2) The Digital ID for Financial Services Working Group (led by the WBG), and
(3) The Security, Infrastructure and Trust Working Group (led by the ITU).

FIGI hosts three annual symposia to assemble national authorities, the private sector, and other relevant stakeholders to share emerging insights from the Working Groups and country level implementation.

This report is a product of the FIGI Security, Infrastructure and Trust Working Group, led by the International Telecommunication Union. The findings, interpretations, and conclusions expressed in this work do not necessarily reflect the views of the Financial Inclusion Global Initiative partners including the Committee on Payments and Market Infrastructures, the Bill & Melinda Gates Foundation, the International Telecommunication Union, or the World Bank (including its Board of Executive Directors or the governments they represent). The mention of specific companies, or of certain manufacturers' products does not imply that they are endorsed nor recommended by ITU in preference to others of a similar nature that are not mentioned. Errors and omissions excepted, the names of proprietary products are distinguished by initial capital letters. The FIGI partners do not guarantee the accuracy of the data included in this work. The boundaries, colours, denominations, and other information shown on any map in this work do not imply any judgment on the part of the FIGI partners concerning the legal status of any country, territory, city or area or of its authorities or the endorsement or acceptance of such boundaries.

About this report

This report was written by Wolfgang Balzer, Focus Infocom and was produced as part of the deliverable of the Quality of Service (QoS) Workstream of the FIGI Security Infrastructure and Trust Working Group. The report is based on a study on QoS field measurements for digital financial services in Ghana, Rwanda and Uganda. The author would like to thank Kwame Baah-Acheamfuor for reviewing the report and facilitating the process for coordinating with the teams in Ghana, Rwanda and Uganda for the QoS field measurements. The author would also like to thank Vijay Mauree and Arnold Kibuuka for reviewing and editing the report and the Security, Infrastructure and Trust Working Group for their feedback.

If you would like to provide any additional information, please contact Vijay Mauree at tsbfigisit@itu.int.

Contents

Executive Summary

The present document describes the QoE assessment methodology for the use case “Person-to-Person" (P2P) money transfer in the cases of person-to-person money transfer in a generalized context which includes inter-operator and cross-border use cases.
This work is based on ITU-T Recommendation G.1033, where a conceptual framework for Quality of service and quality of experience aspects of digital financial services is standardized, and on ITU-T Recommendation P.1502 which standardizes a methodology for QoE testing of digital financial services for person-to-person money transfers for the basic P2P transfers between two devices using the in the same network and DFS operator.
The present document has three main elements. Firstly, the methodological framework and use case definitions for a generalized P2P money transfer use case are given. In this framework, the DFS operator used to send money, and the operator receiving money (i.e., the A and B side of a money transfer) are parameters of the use case, which integrates all variations (same operator/inter-operators; same country/cross-country) into the same methodological context.
The second element of the methodology is a comprehensive framework for data elements and related processing, and check list templates are provided which also help to achieve data quality by giving implicit guidance to field test teams. Together, these components provide the means for operational robustness and a high level of data quality.
The data objects defined here support test planning and management as well as provide the input data foundation for efficient processing of data. Also, guidance is given on how data processing can be done in a consistent and efficient way by using a SQL database.
Last but not least, the methodology introduces a new tool designed to assist field test teams in data collection. This tool (“multi-stopwatch”), conceptually already suggested in ITU-T Recommendation P.1502, is an electronic time-taking tool similar to a stopwatch but supporting testers to record the events within a DFS test case and upload data entered by testers directly to a central location. It eliminates the needs and weaknesses of manual entering time readings and transferring them to post-processing through multiple transformation stages, e.g., from handwritten notes to entries in a spreadsheet. The inherent data consistency provided by this tool translate into significantly lower effort in data preparation, validation and eventual correction, and to respectively higher levels of efficiency and data yield from tests.

Methodology for inter-operator and cross-border P2P money transfers

1 ABOUT THIS DOCUMENT

ITU-T Recommendation P.1502 defines a Methodology for QoE testing of digital financial services for the use case of person-to-person money transfers within the same DFS operator.
The present document extends this methodology towards Inter-operator and cross-country P2P money transfers.
For frame and introductory information, please refer to that document.

Keywords
Digital Financial Services, Methodology, QoE, QoS.

2 INTRODUCTION

In ITU-T Recommendation G.1033[1], a conceptual framework for Quality of service and quality of experience aspects of digital financial services is standardized. ITU-T Recommendation P.1502 then standardizes a methodology for QoE testing of digital financial services for person-to-person money transfers.
As stated in the Introduction of P.1502, it is not expected that there will be a universal QoS and QoE test suite that could be applied to all DFS applications. These Recommendations do, however, provide a solid basis for derived methodologies tailored to specific situations.

The present document is an example of this approach at work. It extends the methodology given in P.1502 to inter-operator and cross-country use cases of P2P money transfer, and it also shows how tool-assisted testing can improve data quality and robustness of testing.

3 SCOPE

The present document describes the QoE assessment methodology for the use case "Person-to-Person" (P2P) money transfer in the cases of person-to-person money transfer in a generalized context which includes inter-operator and cross-border use cases.
As in the case of P.1502, this methodology only covers the methodology for tests done from an individual user's (end to end) perspective, acting within a given DFS ecosystem under current load conditions.

4 REFERENCES

[G.1033] Recommendation ITU-T G.1033 (10/2019), *Quality of service and quality of experience aspects of digital financial services*: https://www.itu.int/rec/T-REC-G.1033/en.
[P.1502] Recommendation ITU-T P.1502 (01/2020), *Methodology for QoE testing of digital financial services*: https://www.itu.int/rec/T-REC-P.1502/en.

[E.840] Recommendation ITU-T E.840 (06/2018), *Statistical framework for end-to-end network performance benchmark scoring and ranking*: https://www.itu.int/rec/T-REC-E.840/en.

[b-ETSI TS 102 250-6] *QoS aspects for popular services in GSM and 3G networks*;
Part 6: Post processing and statistical methods (2004-10), as contained in ITU-T Recommendation E.804 section 11: https://www.itu.int/rec/T-REC-E.804/en.

[b-DFS TR] ITU-T Focus Group Digital Financial Services, Technical Report (05/2016) *QoS and QoE Aspects of Digital Financial Services* FG DFS QoS Report.

[b-FIGI-1] Financial Inclusion Global Initiative (FIGI), Security, Infrastructure and Trust Working Group (SIT WG) (03/2019), *Methodology for measurement of QoS KPIs for DFS* Methodology for measurement of QoS KPIs for DFS.

[b-FIGI-2] Financial Inclusion Global Initiative (FIGI), Security, Infrastructure and Trust Working Group (SIT WG) (03/2019), *Report on the DFS pilot measurement campaign conducted in Ghana* Pilot measurement of QoS KPIs for DFS in Ghana.

[b-FIGI-3] Financial Inclusion Global Initiative (FIGI), Security, Infrastructure and Trust Working Group (SIT WG) (2019), *DFS Consumer Competency Framework.* See also: https://www.itu.int/en/ITU-T/extcoop/figisymposium/Pages/FIGISITWG.aspx.

5 DEFINITIONS

None.

6 ABBREVIATIONS AND ACRONYMS

Please refer to P.1502 for a full list of abbreviations. The following list contains only newly created abbreviations and, for convenience of reading, the most frequently used abbreviations in the context of Digital Financial Services.

API	Application Programming Interface
DAL	Device Assignment List (see Methodology for a full explanation)
DFS	Digital Financial Services
E2E	End-to-end
FTL	Field Test Lead (role name, the person responsible for directing field tests)
ITU-T	International Telecommunication Union, Telecom Standardization sector
KPI	Key Performance Indicator
MSW	Multi-stopwatch tool (see Methodology for a full description)
NSMS	Notification SMS
PIN	Personal Identification Number
P2P	Person-to-Person
QoE	Quality of Experience
QoS	Quality of Service
RAT	Radio Access Technology
TA	Transaction
TAL	Team Assignment List (see Methodology for a full explanation)
SMS	Short Message Service (also used for a single text message transmitted by SMS)

7 CONVENTIONS

The following terms are used in an interchangeable manner:

Working name or definition	Term/Alias
DFS (Digital Financial Services)	MoMo (Mobile Money)
A or B Party, Account (actually the representation of a user's account on a mobile device or another type of TE)	Digital wallet, Wallet
MSW	Multi Stop watch, tool for time-taking of events
TA	Transaction
ObsTool	Observer Tool: User Equipment running software for active and passive network testing in the background

It is important to note that Digital Financial Services in most cases cannot be understood as "standardized services" like telephony or facsimile, but rather as applications having an internal functionality which is not known to the general public and may also change over time without prior notice.

8 TEST SCENARIO UNDER CONSIDERATION

The basic scenario under consideration is the "Person-to-Person" (P2P) money transfer in the cases:

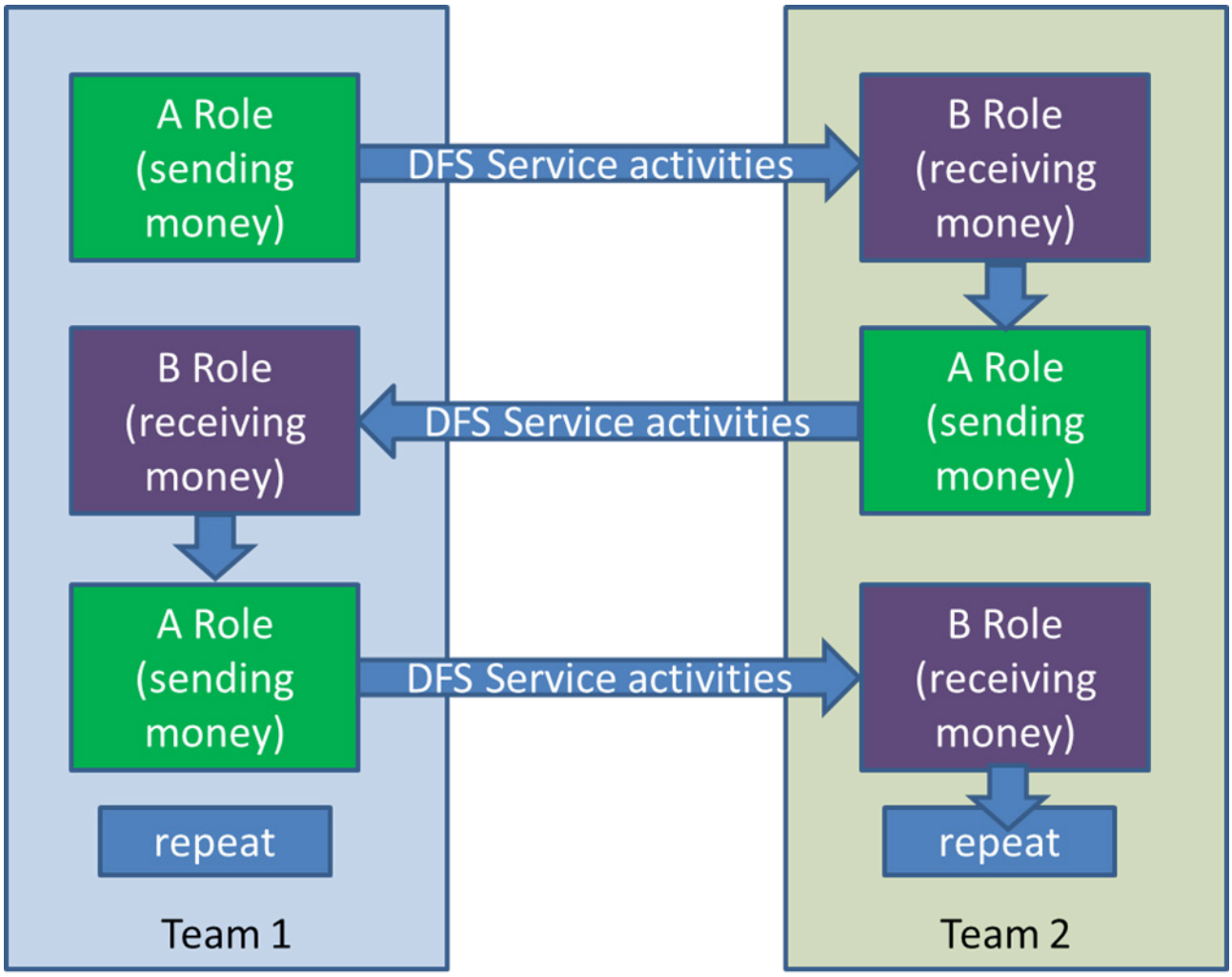
- Money transfer between two parties in the same country, but using different DFS providers (inter-operator scenario).
- Money transfer between two parties in different countries, using the same DFS providers (e.g., national branches of a multi-national network operator).
- Money transfer between two parties in different countries, using different DFS providers.

It is important to state that the methodology for these variants is the same as in all these cases money is transferred between two entities. There may be differences in the details of an operating sequence; these are, however, not greater than differences between same-operator, same-country operating sequences.
NOTE – In some countries, the DFS service might be registered under the central bank, and money transfers considered to be bank transfers. This may have an effect on the appearance of respective transactions where, in actual implementations, it has shown that in some cases entities other than the sending party (e.g., agents) are used for cross-border transactions. From an end to end perspective, this is still a P2P transfer; technically, received money will in this case appear to come from that particular entity rather than from the actual sending entity. If reception notifications are used for data evaluation (which is not the case in the current methodology), this would have to be considered in data processing.
The P2P basic scenario and its modelling is described in detail in ITU-T Recommendation G.1033 and ITU-T Recommendation P.1502. For the sake of convenience of reading, the essential parts are explained here while for more detail, the reader is kindly referred to above mentioned Recommendations.

8.1 Roles, entities and action/event flow

In the P2P money transfer scenario, money is transferred from party A (the active party which is sending money) to party B (the receiving party.
In a practical implementation if testing, each party is represented by one testing team. By practical considerations, money is transferred in a cyclic fashion, so teams switch roles after each transfer, as shown in Figure 1.
NOTE – The graph below only shows the basic case. In order to take care of the whole spectrum of possible cases during testing, some additional, derived cases need to be considered, which will be done in a subsequent section.

Figure 1 – P2P money transfer roles and team activities



8.2 Test parameterization and neutral starting state

A particular property of systematic service tests is a frequency of service usages which is significantly higher than the usage frequency created by a typical end user.

While a high testing frequency leads to a high yield of samples for computation of QoS KPI, it is conceivable that the system has a certain "dead time" after each transaction, where the system would not accept a new transaction or create unexpected results of a transaction attempted within this period of time. It is advisable to be aware of this possibility and obtain respective information before actual parameters of a test campaign are determined.

The testing frequency can be controlled by a pause between transactions, which also acts as a guard time to allow the service under test to reach its neutral state again. Respective considerations are in full analogy to testing of e.g., telephony.

A testing campaign, therefore, should contain a pre-testing phase with systematic tests to make sure that usage frequencies typical for testing do not affect testing results with respect to the end-user perspective.

NOTE – In actual implementations, there may be limitations applied by the DFS providers limiting the number of transactions per day, or the total amount of transferred money per day or another period of time. It is also conceivable that mechanisms exist which limit the frequency of testing. When setting up tests, it is important to check for such conditions. This starts with creating awareness of testers for effects of such mechanisms, and be prepared to adapt data processing accordingly if respective

effects are detected during a testing campaign. Otherwise mis-classifications may result, such as attributing effects of such limitation as functional failures of a MoMo service.
As the starting hypothesis for systematic testing, it is assumed that a guard time is typically in the range of 10 to 30 seconds.
When testing is done manually, it is assumed that the system can handle all testing speeds which can be realized by human testers, as even an experienced tester will not work significantly faster than an experienced regular user of DFS. Therefore, no special requirements to slow down testing are applied.
In fully automated testing, it would also be possible to use the high degree of repeatability of such control to determine the appropriate guard time by probing, i.e., by systematically varying the guard time and check for respective effects.
There is a second category of effects which need to be considered, namely the possibility of a service-specific local memory (analogously to a browser's cache) which stores information related to previous transactions. The effect would be that in subsequent transactions, such information would be read from local memory instead of obtaining them by an over the air request to the service. This could then impact related measurement values or KPI.
As long as effects are quantitative rather than qualitative, it may not practicable and is not necessarily required to exclude frequency-dependent effects entirely. However, respective effects need to be recorded and documented carefully as part of the reporting in order to understand their impact on the testing conditions.

8.3 Re-initialization after unsuccessful transactions

If a transaction fails, in particular after a time-out condition has occurred, it shall be ensured that the service and the device or application are in the typical neutral starting state again, i.e., that no memory of previous error states remains in the system.

8.4 Disappeared Money

It is possible that during a transaction, the amount of money deducted is not correct with respect to transferred amount and fees. This includes the case that the amount is correct but sent to a third party by an error in the system. From an end customer perspective, this is either a loss (if too much money is deducted), or an unjustified gain (if money is credited but not deducted on the other side of the transaction. For simplicity, we use the term "disappear" for both variants of this kind of effect.
There may be undelivered transactions where money is deducted from the sender's account, along with transfer charges. In such cases, it will typically be required to fill a complaint with the MoMo service operator. If this complaint is successful, money will be returned at a later point in time (depending on the process and the MoMo operator's terms of service, transfer charges will not be refunded.
Retrieval of lost money is understood as a second stream of activities outside the scope of this methodology. Functionally, even if money is returned later, it will reduce the available credit for further tests. Therefore, in all cases of disappeared money, insertion of fresh money may be necessary to keep up the necessary level of credit for further testing.
The matter of transaction failures needs special consideration. In that case, it is assumed that a typical user seeks confirmation, by e.g., calling or messaging the recipient (i.e., using an external means of communication). Also, in many cases, the receiver would issue a receipt confirming incoming payments. The sending party might wait for that statement and inquire.
In any case, in particular in testing modes where the A party has no direct visibility of events on the B-party (this issue is also discussed in subsequent sections), reasonable and appropriate measures and conventions, adapted to the actual scope and goals, shall be considered and set-up as part of a testing campaign.

8.6 Automation of tests

The methodology in the present document describes testing in a generic way, i.e., service tests can be done manually as well as in an automated way. It is understood that automation of tests is desirable to achieve a greater degree of repeatability, and less variation in quantitative data values due to inaccuracy of e.g., manual time measurements.
Automation can have different forms with respective degrees of automation up to fully automated testing. Using the multi-stopwatch concept as described in this methodology is the next step of evolution, significantly improving the robustness of testing with respect to manual event logging.
The next step may be to still use manual operation of transactions but to record low-level activities on the DFS device itself, e.g., from recording of Layer 3 messages or IP-level activities. The ultimate goal would be a system which executes the whole MoMo process automatically. Respective implementations require, at least, an extended level of access to platform devices ("rooting", e.g., having system-level access) and substantial technical efforts, in particular

because testing such systems would require actual use of MoMo services. Making these systems fit for unsupervised operation, or enable operation on a larger number of mobile devices types, would further increase the necessary effort.

Design or implementation of further steps of automation, or related evolution of the methodology is, however, not within the scope of the present work.

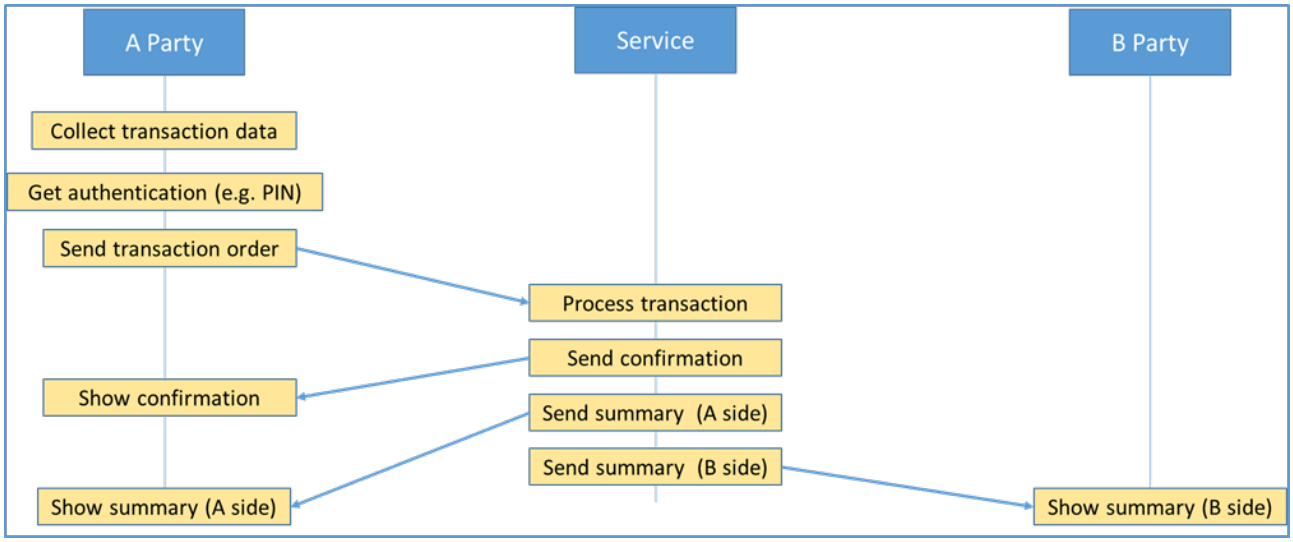
9 TRANSACTION MODEL AND DFS KPI

The basic model is identical to the one described in ITU-T Recommendation P.1502. The following section describes the special variant for inter-operator and cross-border testing where teams do not have direct contact to each other.

In the following, the basic considerations and principal definitions from P.1502, as well as extensions, are described. For a more thorough background reading, as well as for the full set of DFS KPI please refer to P.1502.

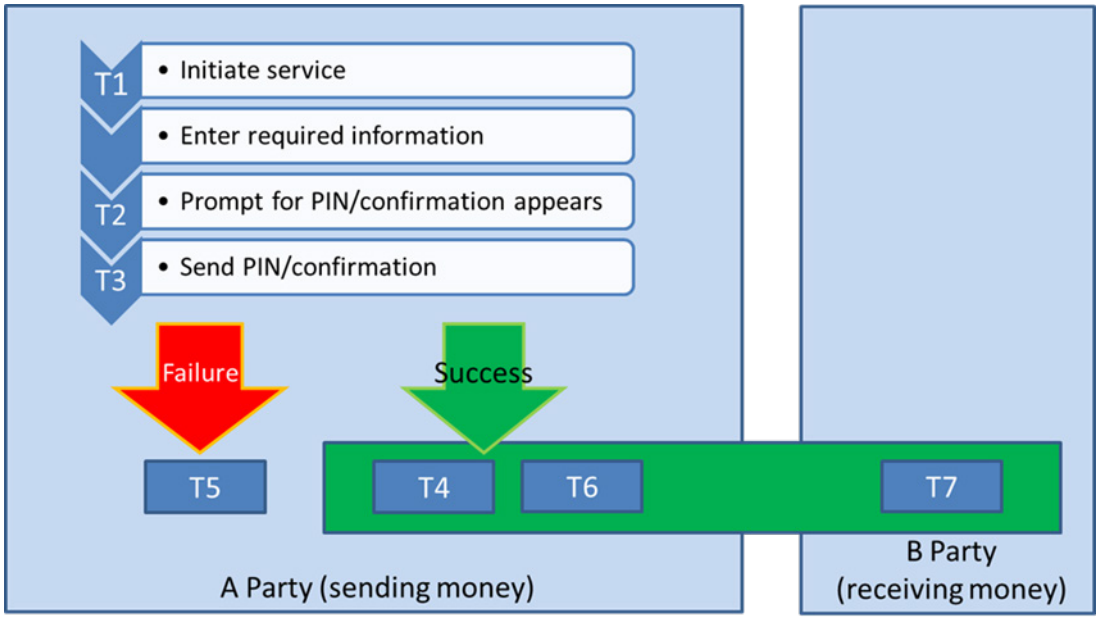
Figure 2 shows the basic structure and event flow of the DFS implementation; the collection of required details ("Collect transaction data") is shown summarily; the details are different between operators.

Figure 2 – Basic model of a P2P money transfer implementation



The following figure show how the action and event flow is mapped to the "timer flag" elements which are then used to compute the KPI. Again, the basic processing is the same as for the basic P2P case.

Figure 3 – Connection between events and timer flags (see text for details)



T4 is the primary (in-application) success criterion while T6 refers to the success criterion provided by reception of a notification SMS. Please note that in this generalized case, T4 and T6 can appear in any combination and order.

The principal difference to an intra-country, intra-network is that most likely teams are working in different locations[2]. Therefore, not all of the timer flags are recorded in the same place. The practical consequence is that data has to be combined from different sources, which is described in more detail in a subsequent section of this document.

It is important to keep in mind that due to the different paths for events T4, T6 and T7, they can appear in any order.

Due to the different types of implementation, it is possible that on the A Party side, either T4 or T6 is missing. For data processing, this means that transaction success is indicated by either criterion.

The KPI used are a subset of the simplified set defined in P.1502 clause 10 (and Table 5 there). Please note that the event names are the ones defined in the present document; the set differs from that in P.1502.

Remark 1: As for MTCR, the set of TA used for computation depends on decisions to be made. Basically, a TA is valid if is not marked as "to be ignored" due to reported input errors, and if it is within the "maximum transaction count" limit as previously described. To be defined: use transactions where network or service failures have been reported?

Table 1 Simplified set of KPI used in the present document, based on the set in P.1502

Indicator	Abbreviation	Definition/Remark
Money Transfer Core Duration [s]	MTCD	T4-T3
Money Transfer Raw Completion Time [s]	MTRCT	T4-T1
Money Transfer completion rate [%]	MTCR	T1 present, T4 present: success (see remark 1)
Money Transfer Full Completion Time [s]	MTFCT	T7-T1: Not used when asynchronous mode is used
Money Transfer A-side Completion Time [s]	MTACT	T6-T1 (see remark 2)

Also, the question if MTCR shall be reported at all needs to be decided, w/r to non-representative nature of testing.

Remark 2: In cases where T6 is used as surrogate for T4 in case T4 is missing, MTACT is typically not computed as there would be partial overlap with MTCD and the sample basis is smaller in any case. If computation of MTACT shall be computed nevertheless, it needs to be made clear in accompanying documentation (project report), that this is a secondary/auxiliary KPI.

Special consideration is required for the case of MTCR, as the reported completion rate is depending on the number of unsuccessful transactions. There might be cases where either the network or the DFS infrastructure is temporarily down, i.e., a part of the service is systematically unavailable. MTCR is therefore not only a technical element, but also a matter of testing perspective and scope of a testing campaign. It needs to be clearly defined and documented how these cases are treated, i.e., if and which transactions shall be removed from the valid set.

In the case of a campaign having explorative character, or seeking a broad perspective, a solution might be to report different variants of MTCR to show the corridor of values, depending on respective decisions.

10 CREATING THE USE CASE MODEL FROM ACTUAL USE CASE EXAMPLES

It has been shown that for creation of the basic model of a DFS transaction, a well-produced video is best practice.

A well-produced video is a persistent source of information in detail and can be analysed easily.

In order to fulfil its purpose, videos should be produced along the following guiding lines:

- Show the device screen in good, uniform lighting and clarity, avoiding light reflections.
- Make sure that while there is of course the need of manual operation, the screen is visible long enough in each step to allow following the flow of events.
- Have a high-quality audio comment providing explanations of the steps to be taken, and comments on results where necessary. Ideally, these comments already include references to actual event-recording processes, e.g., mentioning the "timer flags" to be recorded.
- If feasible and for completeness, the screen of the B party device should also visible. While this will be of course impractical if the B party is in a different location, it may be provided by another video showing the reaction of the device on an incoming DFS transaction.

11 DATA SOURCES

In order to compute DFS KPI, respective input data need to be collected.

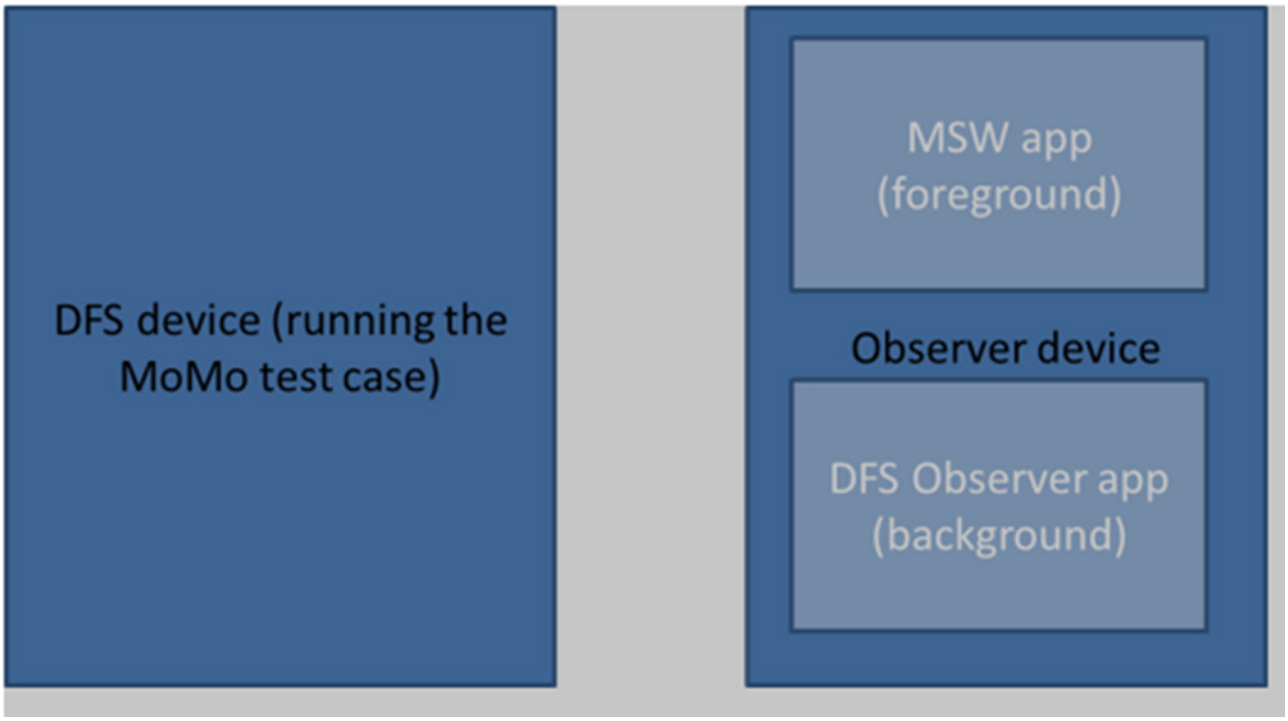
Basically there are three sources of information:

- Recorded events from observation of the DFS use case. In the context of the present document, these are events recorded by the Multi-stopwatch (MSW) app.
- Results from background measurement of the transport network at each side of the DFS use case. Basically this can be done by every suitable QoS testing tool on the market. For the purpose of this document, it is assumed that background testing is represented by the app named "DFS Observer" which was also used in the first Ghana pilot test described in ITU-T Recommendation P.1502.

Optionally, notification SMS on both sides of the DFS use case. In the current context, these are not used as there is considerable effort to collect these data and their additional value is considered to be small. In case it is desired to use them, refer to the respective sections of ITU-T Recommendation P.1502, or the respective FIGI reports (see References).

The data source apps are installed on platform devices. On the Android operating system, running apps concurrently without cross-effects is not guaranteed so the ideal configuration, from a fundamentalist point of view, is to run each app on a separate device. On the other hand, it is desirable to minimize

Figure 4 – Schematic allocation of data sources to devices



the handling effort, i.e., to combine apps on the same platform. The actual decision is made by the Field Test Lead (FTL) on the basis of a benefit-to-effort consideration. It is recommended to also run checks (e.g., on the data from the first days of a measurement campaign) to make sure that there are no negative effects, or only effects on a tolerable scale.
The DFS application must be always visible in order to record events properly, and the MSW app also needs to be on top of the screen for delay-free recording of events. Therefore from all possible combinations, only a few remain: There must be at least two devices; the network-testing app can run on either the device running the DFS use case ("DFS device"), or the device running the MSW app. With the further consideration that network testing also uses packet data resources, the second configuration is the one of choice, as shown schematically in Figure 4.
Please note that this is a rather schematic and simplified view. Details given in the following sub clauses have precedence.

11.1 Basic Considerations for data collection and processing

In the case of measurement campaigns involving cross-border transactions, and transactions between different networks, one or several of the following conditions can apply:

- Testing teams may be in different locations without direct communication between them. This means that each team only sees a part of the overall set of events belonging to a transaction. NOTE – Enabling real-time communication between teams – e.g., via audio/video conferencing solutions) requires adequate data bandwidth which limits applicability to respective situations. A powerful means of communication between teams would be an automated dashboard, making processed information available in near real time. Technically, such a solution is easily feasible given the required amount of budget. In the actual design of such a system, using as little as possible data bandwidth would have to be a major design goal in order to assure operation under a wide range of on-site conditions.
- There may be multiple teams at work in the same time period.

Consequently, additional information – beyond what is collected by the testing tools itself – is required to produce the desired information, i.e., DFS KPI.
Basically, it must be known which team or pair of teams is running which scenario in which period of

time. Only then it is possible to create respective KPI correctly.
For instance, assume Team 1 and 2 are running a national test between two DFS providers, and at the same time, team 3 and 4 is running a cross-border test on another set of DFS providers. For a given transaction for the national test, timer flags T1 to T6, are taken by team 1, while T7 is taken by team 2. For the cross-border test, T1 to T6 is taken by team 3 and T7 is taken by T7. Likewise, network performance tests are also collected by respective devices in different locations.
Assuming that all data is imported to the same database – which is the usual way to process data for best efficiency – allocation of respective device ID to DFS service test scenarios has to be made, which requires corresponding information about measurement system allocation and testing schedules. Also, the process of data cleansing – removing of data which is considered to be not valid – requires respective information.
In many cases, it may be possible to deduct information which identifies the scenario under test by using information in the primary data source, e.g., GPS locations in the background-testing data. However, as the methodology is supposed to work in a robust way under a wide range of conditions, it cannot be guaranteed that these information sources will always have sufficient information. For instance, if tests are done from within a building, a valid GPS position fix may not be available. Therefore, frame or top-level information should be provided.
Basically, such frame information can be provided centrally or locally. As such tests are typically done in a planned manner with a central management entity, there should be a register of team, device, and scenario allocations versus time. For maximum robustness, it is recommended to also collect this information again locally, i.e., using log sheets listing what has actually been done by the teams. As this information is typically high-level, i.e., taken only once per location or testing session, load on teams is low and the extra redundancy improves the overall robustness of testing.
This methodology is designed to be robust in the sense that a certain degree of redundancy is maintained for information which is essential to proper data evaluation. The most essential information is which teams are paired for a given test case, and about the assignment of electronic ID's of respective tool installations.
These electronic ID's are the most essential single elements of a test and measurement set-up, as they are required to assign measurement data to the right context.
Basically there are several types of electronic IDs:

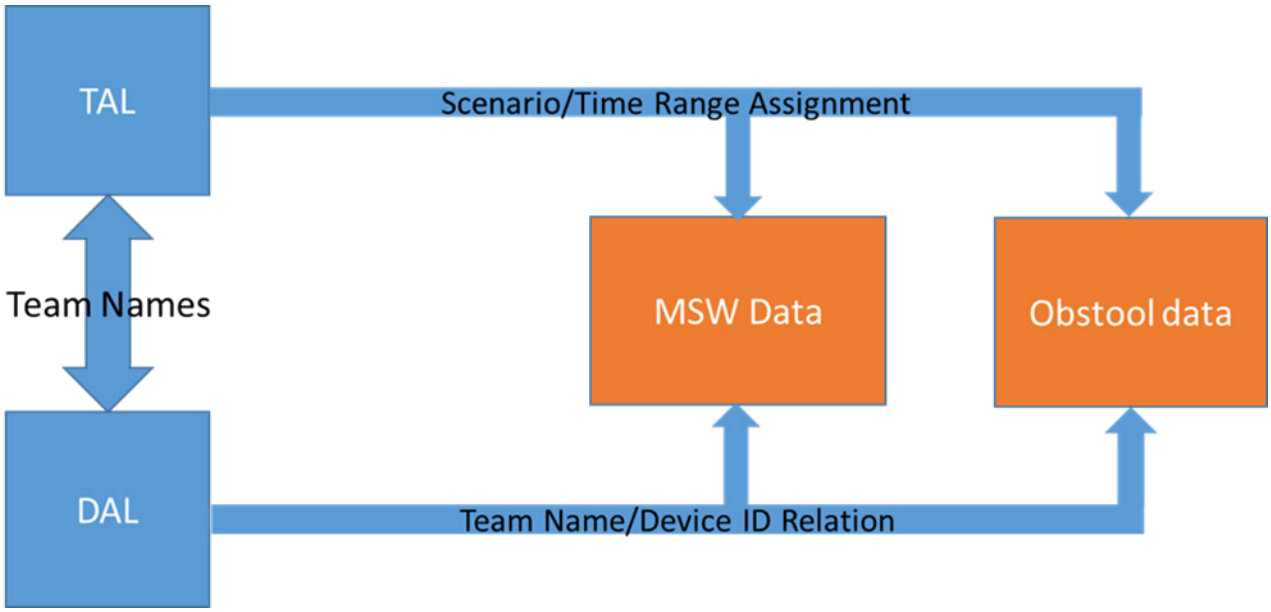
- Fixed ID's, such as IMEI or MAC addresses. In most cases they can be read electronically by apps through respective API's. However, in recent years operating systems put restrictions on this type of ID as they allow identification of devices and therefore – in case where devices are also used for personal purposes – may create privacy issues.
- Dynamic ID's which are created with every new installation and which are not lined to any static attributes or properties of the platform.

In any case, suitable ID's need to be unique, i.e., it must be made sure that no two devices or rather, data sources, have the same ID to prevent mix-up data assignments.
In the current case, it is assumed that devices are sourced exclusively for testing purposes, so fixed ID's are not problematic. The MSW provides, however, dynamic ID generation and is therefore more versatile.
There are two central lists, maintained by the Field Test Lead (FTL) in co-operation with the entity managing the overall testing activities:

- The Device Assignment List (DAL) holds information about the assignment of devices and tool app ID's to teams.
- The Team Assignment List (TAL) is holds information about the time schedule of scenarios and respective team pairings (under the assumption that each team runs one end of a two-way MoMo use case as defined in the Transaction Model). The TAL is used both for planning activities, as for documenting them afterwards.

Important note: If team assignments change, it is especially important to keep good record of these assignments, to prevent data loss or artefacts due to unclear or incorrect combination of data from different teams/data sources.
Copies of the DAL and TAL are imported to the post-processing database and serve as the source of assignment operations required to generate KPI and other report information.
In addition, each team uses log sheets (in paper or electronic form) to document their activities locally. These log sheets provide an additional layer of robustness by providing redundancy of information with respect to central lists.

Figure 5 – Symbolic overview of data structure for post-processing



The lists are described in detail in sections Local log sheets, Team Assignment List (TAL) and Device Assignment List (DAL).

11.2 Data Structure Overview

The overall data structure is shown in Figure 5. It is assumed that this data structure will exist in a central data processing environment, typically a SQL database.

It needs to pointed out that the methodology does not provide a single, prescribed data structure. Actual data structures can have additional members and shapes. Also, there is no absolute way to process data; the methodology can therefore be embedded in a wide range of post-processing environments and tool chains. This applies, in particular, to background testing (network KPI) data which can be created in multiple ways.

Data will be typically be processed in steps which also involve data validation and inspection. The goal is in any case to obtain a robust database for subsequent processing, i.e., any ambiguities, missing assignments or contradictions should be detected and resolved prior to creation of actual deliverable output.

Data preparation and validation is usually a multi-step process starting with coarse "data cleansing" on input data basis (e.g., visual inspection of data in Excel® files and alignment with log data).

As a database is an efficient environment for data inspection and structural checks, data cleansing is typically a cyclical, incremental process.

Examples for data which may need to be cleaned out are:

- Data resulting from test runs which have not been done under defined conditions.
- Data taken in situations which are deemed to be exceptional and should not be part of statistics.
- Data resulting from unintended operation, e.g. a wrong PIN, or from an operation cancelled due to some other wrong entry.

Pre-cleaned data is imported to the database, checked, corrected in case errors are detected, and imported again until the desired state is reached. During this process, individual data sets may be "masked out", i.e., tagged as to be ignored during further processing steps. This is necessary if information is incomplete or contradictory due to missing or inconsistent data collected in the field[3].

Data cleansing may be a cyclical, repetitive process because in order to detect some artefacts a certain level of cleanliness is required in the first place. Also, when it comes to processing larger amounts of data, data need to have some formal structure before meaningful checking procedures can be applied efficiently.

11.3 Naming and formatting conventions

The following conventions are essential to ensure error-free and efficient data processing over the whole chain.

11.3.1 Team Naming

For data processing, consistent naming over all data sources of primary importance as it is used to combine data, and errors will lead to incorrect KPI evaluation.
The following assumptions/prescriptions are made:

- A **DFS operator** is identified by, in this order, the **network name and the country name**. Examples: MTN Rwanda; Airtel Uganda; Vodafone Ghana.
- Team names shall be chosen accordingly, i.e., the team name should be the same as the operator name.
- A team operates a "logical group" of one DFS device and one Observer device (see Test scenario under consideration).
- There are only two permitted types of device/function allocation:

  a) One device is fixedly assigned to the DFS role (i.e., is set up for a particular DFS operator), and the second device is fixedly set up for the Observer role.
  b) Both devices are set up for both roles, but **the DFS operator on both devices is the same**.

  If configuration b) is used, data taken by the MSW and network performance background testing are treated as equivalent with respect to the generating devices.

The word "team" has historic origins. In consequence, the team name represents, and stands for, a particular DFS operator. In this sense "team" is a logical entity rather than meaning a group of people.
Therefore, a team can be represented by a single person. It is possible that a single person or a group of persons operates multiple devices, or switches between different set-ups. In that case, the documentation described in subsequent sections (TAL) needs to list each configuration where the DFS operator changes as a separate team with the respective time window information.
This is because there must be absolute clarity (i.e., at all times) in the relation between the Observer device/data ID's (multistopwatch and background testing) and the DFS operator under test.

11.4 Naming and formatting conventions for data objects

In order to facilitate efficient and error-free transfer of data between file media and databases, the following conventions must be met:

- Column names in data files (e.g., Excel®) shall only contain alphanumeric characters; shall not begin with a numeric character, and shall not contain whitespace characters.
- Date and time shall be given in one of the standard formats. Formats shall not be mixed within the same column.
- In particular for time information, data integrity has to be maintained over the whole processing chain. In particular, this applies to decimal point/comma and delimiters in general. Also, it is known that millisecond formats can be troublesome. It is highly recommended to run tests before going into full-size data processing.
- A proven method to enhance conversion robustness is using "decorated" elements. For instance, if a timestamp shall be given with milliseconds, in can be written with a preceding fixed-size string. In that case, it will be imported as a string (e.g., nvarchar) to the database. Once in the database, substring/cast/parsing operations can be used to convert this element to a datetime element again without the risk of loss of information.

11.5 Local log sheets

There are two variants of log sheets. The FTL assesses the given situation and decides which variant is to be used, and which information is pre-printed and which has to be entered by the teams. This decision is based on the level of skill and experience of teams, and expected frame conditions in a given location.

- A rather explicit Location Log Sheet which has fields for basic scenario and set-up related information, and provides a detailed check list for elements to be checked periodically (on a prox. 2-hourly schedule).
- A short-form Session/Location Log Sheet which also collects information essential for measurement data allocation, but has a simplified status check section which just asks for a general confirmation that operating conditions are still within valid parameters.

It is good practice to use actual file names which contain a reference to the project or campaign they

belong to, some working text which describes their function, and a revision number to support document maintenance and evolution.
Example for full Location Log Sheet: Annex 1
Example for short form Location Log Sheet: Annex 2
These templates can be used electronically, i.e., for data entries into respective file copies on a computer, or by printing from templates and filling out these copies by hand.
In the case of electronic input, files should be copy-protected after completion, and a copy being kept in a safe location, as a precaution for the case of data loss or accidental alteration of content in later stages of processing.
The first type of Log Sheet, providing more guidance but also asking for a higher effort to be filled in, is meant to be used by teams which have limited experience or are meant to work under conditions with a higher level of potential distraction where formalized check lists can provide additional operational robustness. The second type of log sheet is designed for experienced teams, or teams which have to operate under conditions where time to fill in details is limited. In practical situations, it would be e.g., the FTL's task to select the most appropriate type with respect to the actual situation.
The log sheets make provisions for some elements of the testing procedure which are optional, to be decided by the FTL:

- Photographing/scanning log sheets and sending them by e-mail (applicable to cases where printed-out copies are used): Frame information should be available at the point of data evaluation as fast as possible, to allow for further steps in post processing and measurement data quality management. Taking a scan or photo and sending them immediately after a testing session at a given location is a means to that end; it also is a precaution against potential loss of such sheets along the way. If it can be reasonably expected that log sheets will reach the point where they are collected and processed further in short time and with low probability of loss, this step can be omitted. In that case, log sheets with respective fields removed or marked as "not applicable" should be created from the respective templates.
- Entering check marks for intermediate tests of network connectivity or power supply status: If it can be expected that teams have the required skill level and proficiency, log sheets with respective fields removed or marked as "not applicable" should be created from the respective templates.

Another practical simplification can be made by printing log sheet copies with field already pre-set. Respective decisions should be made by the FTL based upon judgement of the actual situation.

11.6 Team Assignment List (TAL)

Remark: This type of list is also used with the name Scenario Master List.
This list holds the information about the assignment between team names, scenarios, and time windows associated to scenarios.
In data processing, it is used – in combination with the DAL (see next subsection) to link measurement data with respective scenarios.
The TAL structure is chosen such that it is user-friendly (in the sense that its structure is not more formal than required). However, it must be clearly emphasized that great care is strongly advised when creating and maintaining this list. Undetected errors can cause artefacts and errors in later stages of processing which are hard to detect. In particular, the naming of teams and configurations must be consistent between the DAL and the TAL.
Assuming that DAL and TAL are finally imported to a data base for final processing, it is highly recommended to include respective check and validation procedures for TAL and DAL structure and content.
As the content of a TAL is quite project-specific, no templates are provided. It is however recommended to use the table below for construction of respective files.

Table 2 Structure of a typical Team Assignment Table (TAL)

Column name	Example	Type	Function	Hints/Further remarks
Test_Scenario	InterNetwork	Text	Scenario description	See further remarks
OwnerTeam	MTN Ghana	Text	Team name	Can be set by Excel formula (default: equal to FromConfigName)
FromCountry	Ghana	Text	Country ("From" role)	Primary input field
FromOperator	MTN	Text	DFS/network operator, "From" role	Primary input field
ToCountry	Ghana	Text	Country ("To" role)	
ToOperator	Vodafone	Text	DFS/network operator, "To" role	
Status	Completed	Text	Status of tests	As the TAL can also serve as planning element, the status may contain different states such as "Ongoing" or "Cancelled"
Remarks	Unlimited transfers	Text	Additional scenario-specific information	
MaxTAPerDay	10	Integer	Indicates if there is a systematic limit for the maximum number of transactions per day	Leave this field empty if there is no such limit. The value is intended to be used to compute mask-out indicators during data evaluation. Enter only values which affect the number of outgoing transactions
From_Date_Time	11.05.2020 00:00	Date (time optional)	Start of assignment	
To_Date_Time	15.05.2020 23:59	Date/time	End of assignment (inclusive)	Make sure to include time for proper time windowing
FromConfigName	MTN Ghana	Text	Configuration name, "from" role"	Typically constructed from respective operator/country fields by formula
ToConfigName	Vodafone Ghana	Text	Configuration name, "to" role"	Typically constructed from respective operator/country fields by formula
TransactFlow	MTN Ghana to Vodafone Ghana	Text	Auxiliary field for readability	Default: constructed from respective ConfigName fields, can be overwritten

Remarks:

- This list is understood to describe the minimum required for processing. The list can be amended by additional columns of seen fit by campaign management.
- The TAL uses a couple of primary input fields (e.g., for Operator and Country). This facilitates generation of "logical names" needed for processing in an automated way, e.g., by Excel® formulas. Experience has shown that when entering free text, errors which are hard to see by eye can easily slip in (e.g., two blanks instead of one). These errors are hard to see with the human exe but can cause substantial trouble in automated processing. Therefore it is recommended to use formulae by default and only override content when strictly necessary.
- Some of the fields (e.g., TransactFlow) are there for convenience of usage. If the TAL is an Excel® or similar list, these fields should also be pre-set with respective formulae, to be overwritten when appropriate.
- Please note that the TAL uses the Configuration names (not the Team names) as the purpose of the TAL is to link measurement data to scenarios.
- For final processing and computation, a unique scenario descriptor is required (typically, in databases aggregation of data into respective KPI is done using GROUP statements). This can be done

directly using the Scenario description field in the TAL. In that case, it must be made sure that each description text is unique. Another way can be to use a generic Scenario descriptor (e.g., for a category such as "Cross-border" and to construct the final scenario description use for aggregation from respective fields of the TAL, e.g., from the Scenario, the FromConfigName, and the ToConfigName.
- Date range information may be crucial for later allocation of data. Typically, JOIN statements with corresponding ON conditions including time-range BETWEEN clauses will be used. Make sure that date/time ranges given in the TAL are consistent, either by using complete date/time information, or by using respective DATEADD statements.
- Column names should be chosen to not contain blank spaces or other non-alphanumerical characters. This may ease data processing e.g., in SQL data bases. If desired, creation of "friendly text" for output can be done in respective SQL statements, e.g., replacing underscore (_) by blanks.

11.7 Device Assignment List (DAL)

This list connects measurement data to teams.
In the following, we will assume that there are two ID's:

- The MSW ID identifies data from the Multi-stopwatch app.
- Data from network background measurement are identified by the IMEI of the device on which the respective testing app is installed[4].

The DAL structure is chosen such that it is user-friendly (in the sense that its structure is not more formal than required). However, it must be clearly emphasized that great care is strongly advised when creating and maintaining this list. Undetected errors can cause artefacts and errors in later stages of processing which are hard to detect. For instance, a typing error in an IMEI may lead to "ignored" data items with respective consequences for completeness and correctness of results. The same goes for date and time entries.
The basic structure of the DAL mirrors a team, named after the DFS operator and the country it is testing. This team uses a pair of devices according to the basic role assignment and set-up described in previous sections of this document. However, the structure also provides for extensions of this basic scheme:

- A team may change the devices during the campaign, or a re-installation of apps can lead to new ID's being allocated.
- Devices can assume different roles, e.g., devices can switch between the DFS and the Observer role.
- Devices can be swapped between teams, or teams can be allocated to different testing tasks (e.g., a team testing operator A can start to test operator B after some time).

The DAL has a column structure which is, for reading convenience, shown there transposed with explanations for each column.
As the content of a DAL is quite project-specific, no templates are provided. It is however recommended to use the tables below for construction of respective files.

Table 3 Structure and content example of a Device Assignment List (DAL)

Column name	Example	Type	Function	Hints/Further remarks
CheckStatus	Ok	Text	Can be used for validation vs. log files	Free text, can be empty
OwnerTeam	MTN Ghana	Text	Name of the team	By default, same as configuration name (An Excel DAL can use a formula here)
Country	Ghana	Text	Country	
Operator	MTN	Text	Operator (DFS/mobile network)	
ConfigName	MTN Ghana	Text	Logical configuration name (default: Operator<blank>Country)	An Excel DAL can use a formula here
Start_Allocation	30.04.2020 00:00	Date/Time	Date (time is optional)	see remarks
End_Allocation	19.06.2020 00:00	Date/Time	Date/time (see remarks). Can be left empty	see remarks
Config1_DFS_IMEI	354481115999999	Text	IMEI of the device in the primary DFS role	Although an IMEI only contains numeric characters, respective fields should be treated as text to avoid artefacts by conversion. May also use "text-decorated" variants to ease import to databases
Config1_Obstool_IMEI	354481115888888	Text	IMEI of the device in the primary "observer" role	
Config1_FIMSW_ID	c07eead8-62e8-41dc-b3fb-d5c-1b3a7cde6	Text	MSW ID of the device in the primary Observer role	
Config2_DFS_IMEI	354481115888888	Text	IMEI of the device in the secondary DFS role	An Excel DAL can use a formula here to get content from "opposite" fields
Config2_Obstool_IMEI	354481115999999	Text	IMEI of the device in the secondary "observer" role	
Config2_FIMSW_ID	d12c73fe-8961-450a-8b9f-bc431e394c3e	Text	MSW ID of the device in the secondary Observer role	
Remarks		Text	Free text	

Remarks:

- The DAL structure shown here has the functionally required minimum number of elements. Additional elements may be useful, such as e.g., the phone number associated to a device.
- Typically, the Start_Allocation and End_Allocation fields are including pre- and after-campaign ranges, and actual time windowing for KPI reporting is made in respective processing steps. In a typical data base based post processing, JOIN operations also using these time ranges are applied. These fields can however also be used for a tighter time-windowing only considering actual testing-campaign times.
- If device allocations do not change over time, both the Start_Allocation and End_Allocation fields could be left empty. In that case, data base procedures must make sure proper treatment of content e.g., by setting default dates/times.
- If device allocation changes during the campaign, Start_Allocation and End_Allocation must be used to properly describe these allocations. In respective JOIN operations, SQL BETWEEN statements will be used; if only a date is given w/o time information, 00:00 is assumed. Make sure to that respective time regions are complete (i.e., include

hour/minute information, e.g., 1.2.2020 23:59 in End_Allocation to cover the whole day of 1.2.2020), or use respective DATEADD functions in BETWEEN statements.
- Column names should be chosen to not contain blank spaces or other non-alphanumerical characters. This may ease data processing e.g., in SQL data bases. If desired, creation of "friendly text" for output can be done in respective SQL statements, e.g., replacing underscore (_) by blanks.

12 SPECIAL PROCEDURES IN THE FIELD

12.1 Operational protocol if A and B party do not communicate directly

If teams are operating in different locations, a robust protocol is required which tells teams how and when to act.

This protocol needs to cover the following situations (based on cyclical role-switching in the DFS use case):

- Normal operation: Teams need to start the procedure by agreeing which team is having the initial A-Party role.
- If a team is currently having the B-Party role, and T7 occurs; how long should they wait until continuing in the A-party role? This issue becomes important when it is possible that T7 comes earlier than T4 or T6.
- If a team is in the B-Party role, it is possible the transaction was successful but no notification SMS is received (i.e., no T7 can be set). How should teams act if expected events are overdue?
- How should teams communicate when they want to pause or end testing?

The following figures show graphical representations of the main cases and recommended parameters of the communication protocol.

Figure 6 – Team interaction – normal case

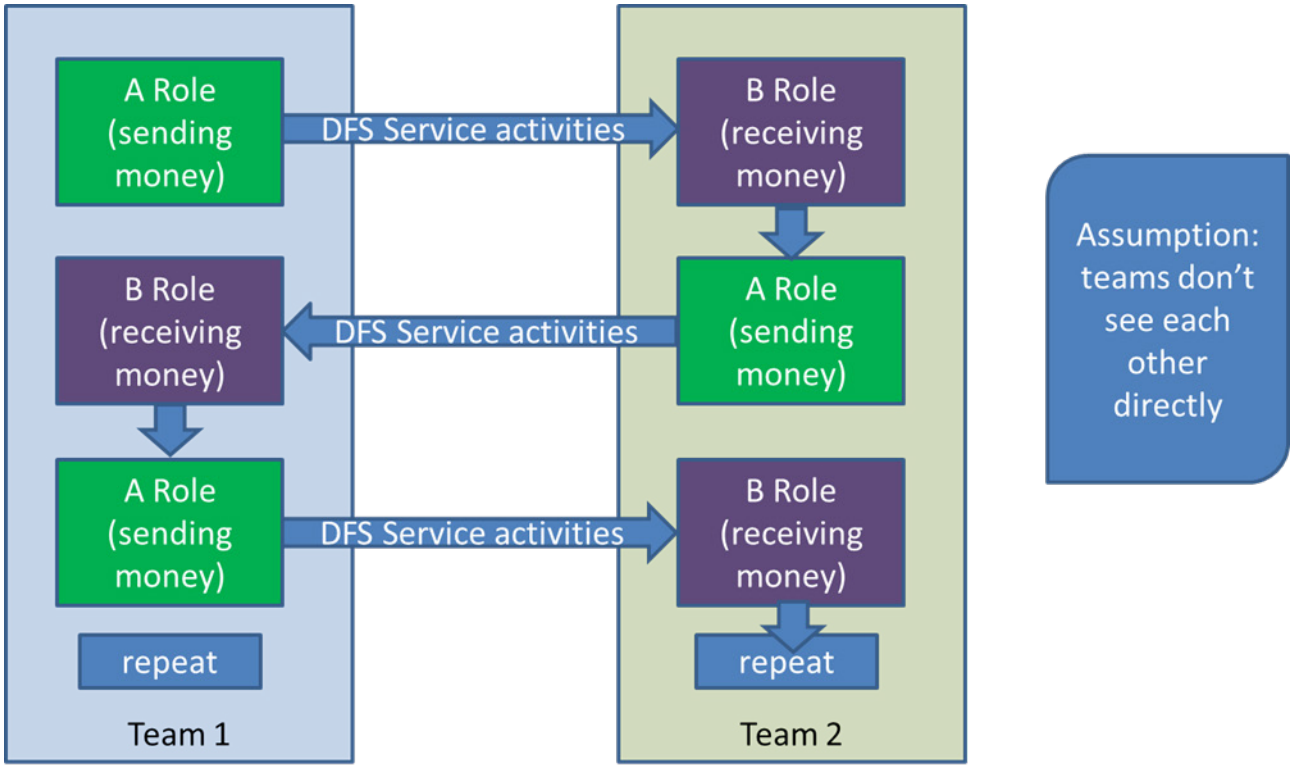


Figure 7 – Team interaction – unplanned interruptions

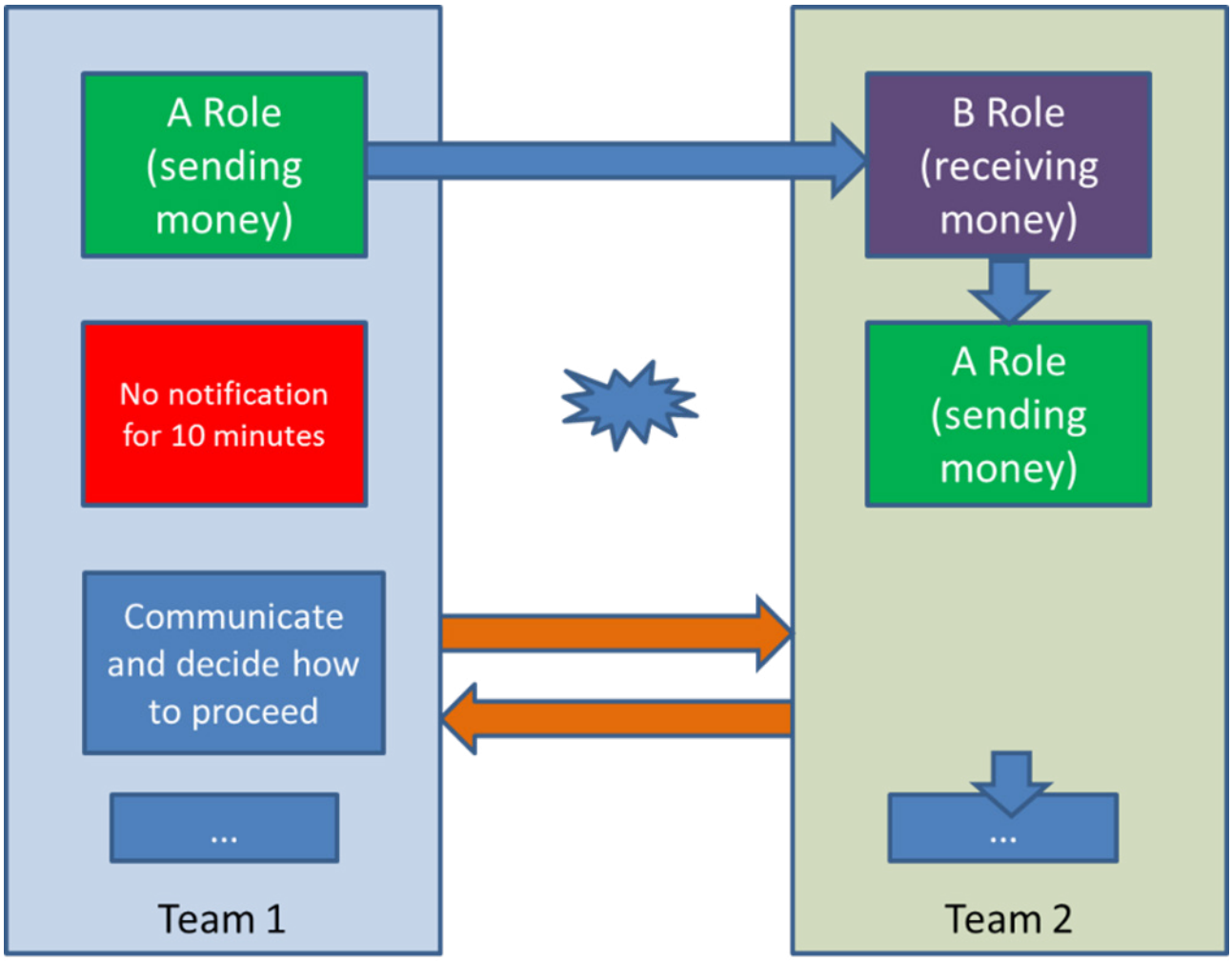
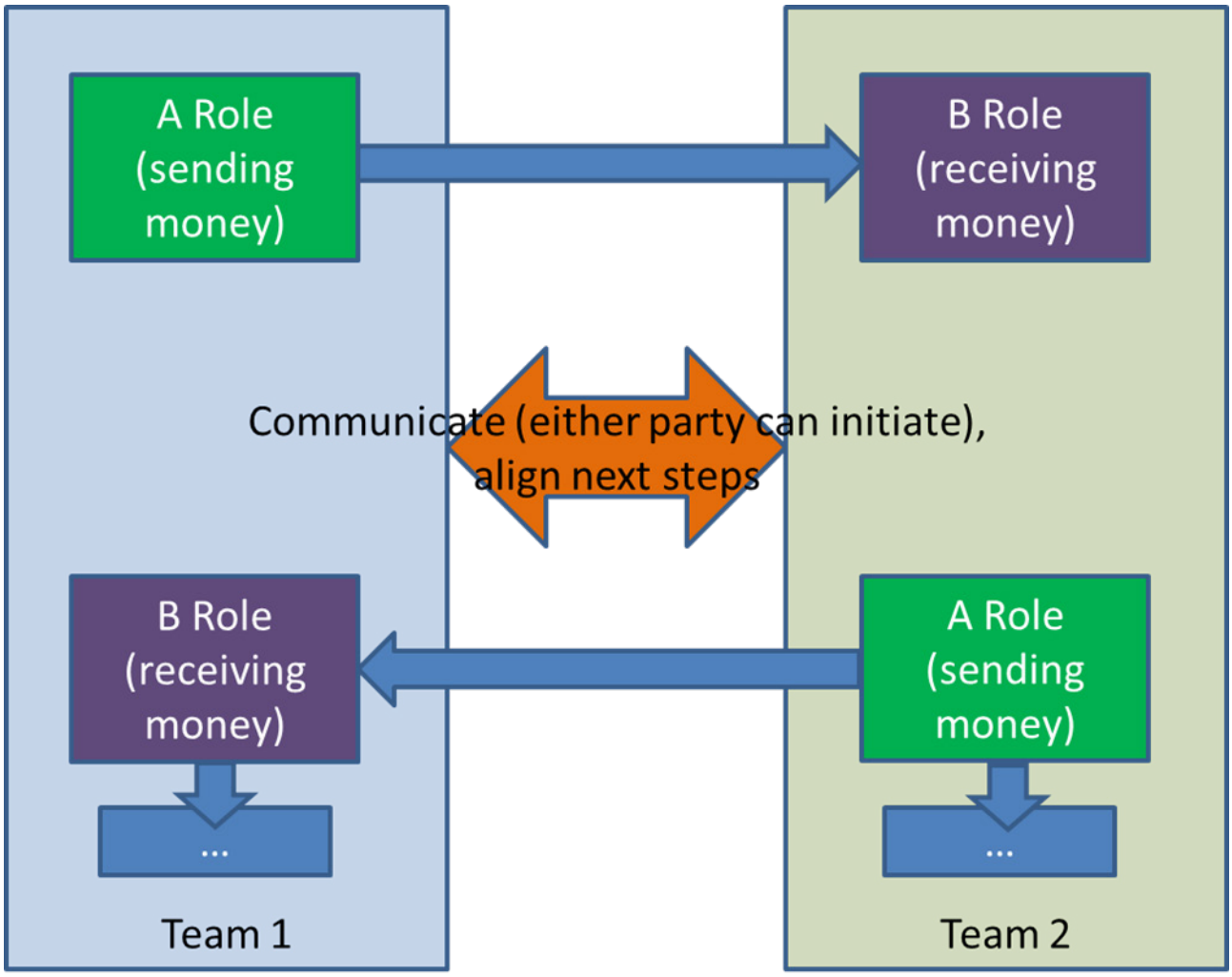


Figure 8 – Team interaction – planned interruptions



13 TESTING MODES

13.1 Basics

In Rec. P.1502, it is assumed that testers have both the A and B Party of a test case configuration under direct control and in full visibility. There is however a number of situations where this is not the case, e.g.

- In cross-border testing where teams are in different locations/countries by nature of the test case.
- In testing situations where special aspects of the service are under test, e.g., with one party in a location with good network coverage, and the other party in a location with poor network coverage.
- In special circumstances as e.g., in a lockdown situation during a pandemic.

In the situation, further difference is made from the degree teams can communicate with each other. A situation where there is full-scale real-time communication, including video feeds on respective devices is practically equal to the standard situation of all elements being in the same room. On the opposite end of the scale, there may be no communication at all between the testers.

13.2 Synchronized and asynchronous testing mode

The following variants of testing, and respective terminology, are defined to take care of these situations. The FTL may select from these variants according to his assessment of the situation and the requirements of KPI to be produced.

Both modes are still assuming paired teams sending money to each other.

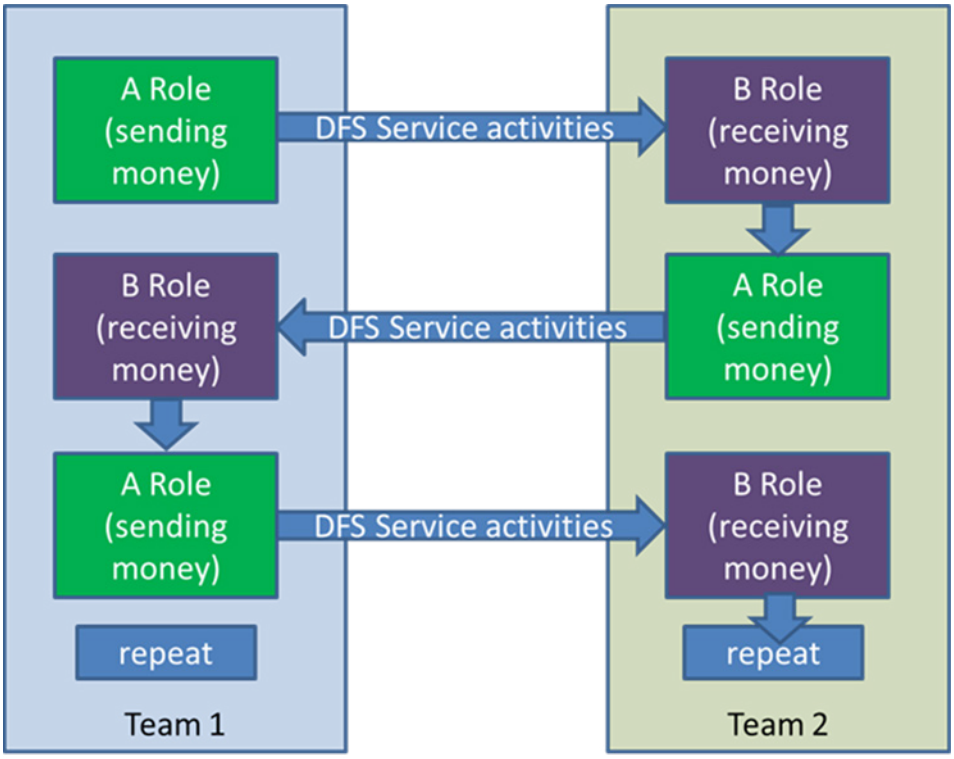
Synchronized testing: In this mode, A and B party roles are exchanged cyclically. The B-party role is ended with reception of a T7 event, or when a pre-defined time-out is reached.

In situations where teams are working in the same location or have real-time communication, one of the testers records events, and a data set can contain all events (T1 to T7) defined in the modelling of the use case.

In situations where teams are separated, the team in A Party role records events T1 through T6 in one data set, and the team in the B Party role records T7 in another one (optionally, T1 can also be recorded in this set, indicating the point in time from where reception of a notification is expected). Respective data are then combined during post processing.

The principal flow in synchronized testing is shown in the figure below.

Figure 9 – Principal action flow for synchronized testing

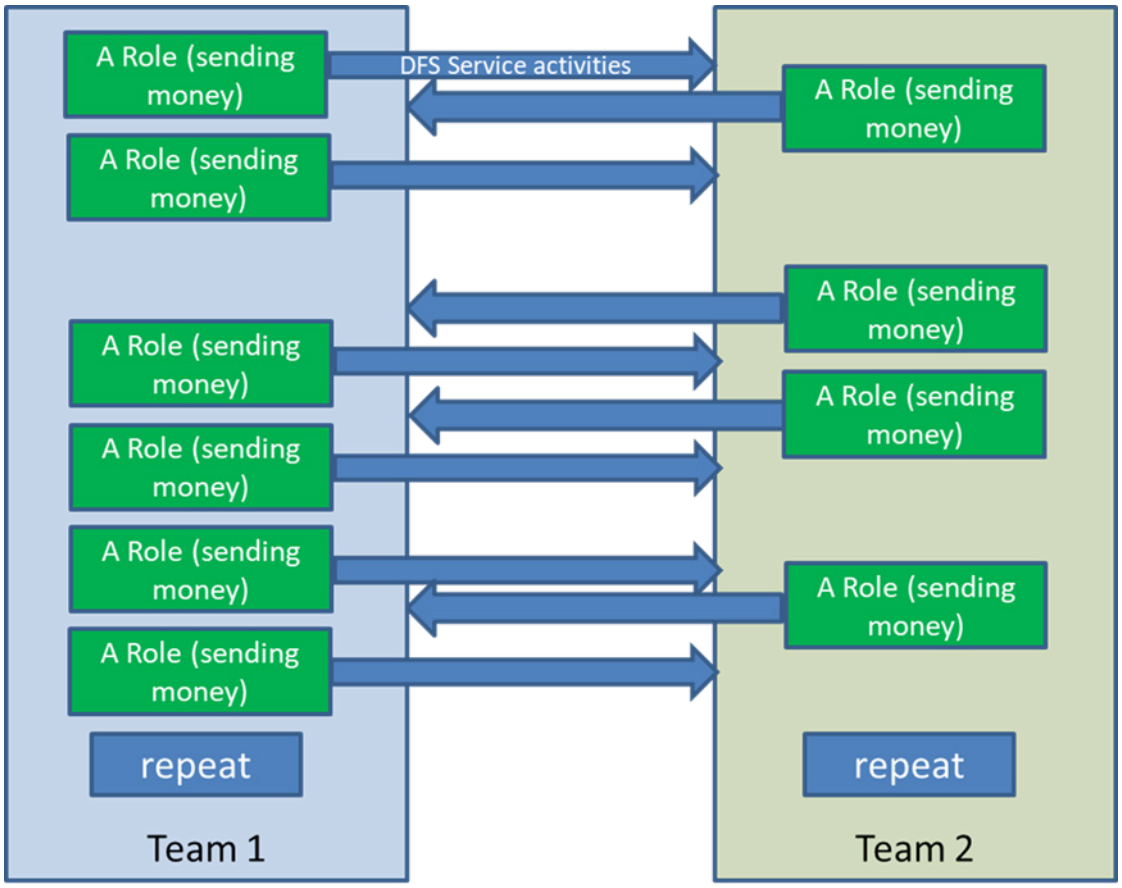


Asynchronous testing: In this mode, testers on both teams act independently in the respective A Party role. This means that Team 1 and Team 2 generate transactions independently towards each other. In this mode, T7 is not recorded, and KPI containing T7 will not be computed and reported.

The asynchronous testing makes sense where teams have different working hours, frequency of operation, or where KPI using T7 are considered to be not required.

In comparison to synchronous testing, the amount of money to be held in credit on each device has to be higher, as in the extreme, only one team may send money for an extended period of time.

Figure 10 – Principal action flow for asynchronous testing



13.3 Dual-function set-up and operation

There is a special situation when intra-operator tests in the same country are made in the same location, and the number of devices used in testing shall be minimized.

In the standard case, devices have fixed functions; the DFS device is used to run the use case while the Observer device is used to run the MSW app and the background-testing app. In the generic case, this would require a total of 4 devices where each device has a dedicated role.

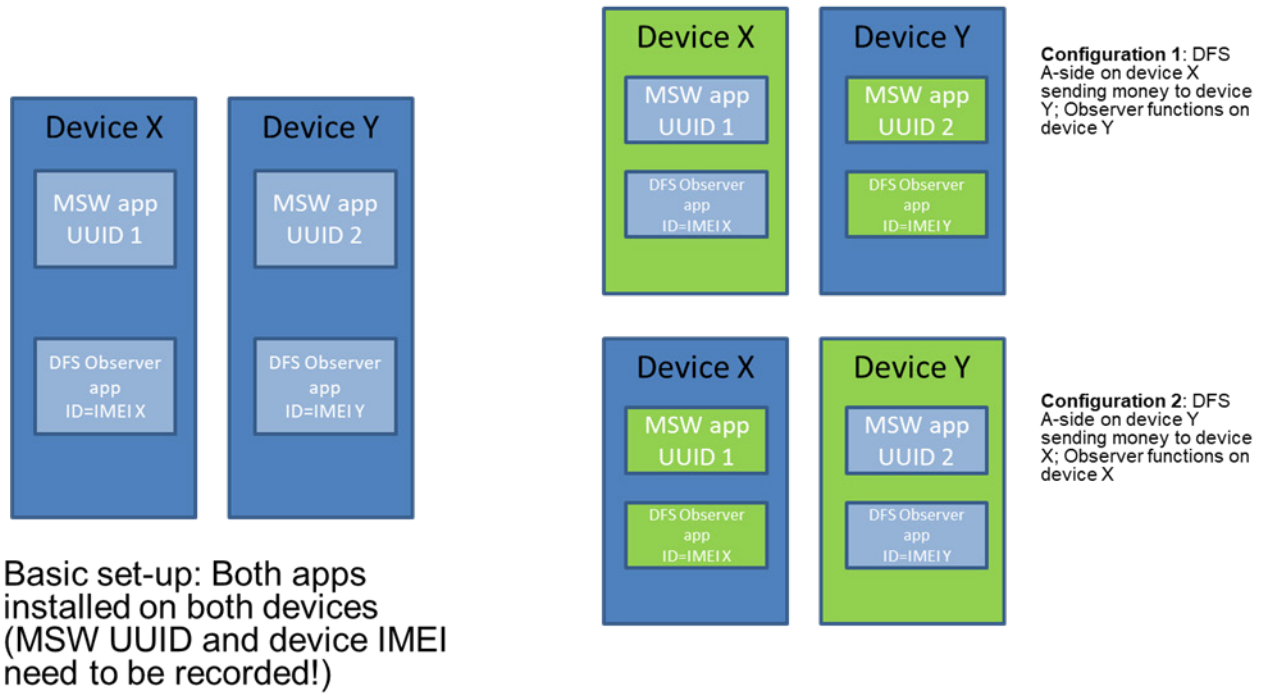
The number of devices can be reduced to two if devices switch functions. In that case, both devices need MoMo accounts, MSW and ObsTool apps installed. Assuming device designations X and Y, the sequence of testing would be:

- X is used as the DFS device; MSW and ObsTool are started on device Y.
- X is in the A Party role, sending money to device Y which is therefore in the B Party role. Events are recorded using the MSW app on device Y.
- If synchronous mode is used, device functions are swapped now. In asynchronous testing, first a number to transactions is run, then functions are swapped.
- After swapping, MSW and DFS Observer is ended on device Y, and started on device X (1). The DFS A-Party role is now performed on device Y with device X in the B-Party role.

Remark (1): This is a precaution to exclude cross-effects from background testing into DFS performance. By assessment and decision of the FTL, respective apps can run continuously on both devices.

The following figure shows the set-up and operation graphically.

Figure 11 – Set-up and operation for dual-function operation of devices



14 RECORDING EVENTS OF THE DFS TEST CASE (MULTI-STOPWATCH APPLICATION)

14.1 Basic Functionality

As already noted in Rec. P.1502, manual collection of test case data requires considerable effort during the tests, and is also a major source of errors and potential degradation of data quality due to transfer of handwritten notes into electronic media.

Therefore, the present methodology assumes there is an electronic tool which supports direct collection of events and related timestamps.

In order to enable independent implementations, this methodology does not prescribe a particular implementation. It does however use terms related to a reference implementation in order to provide a comprehensive picture of functionality, operation and practical considerations related to such an application.

The implementation is based on functional requirements and design considerations:

- The app provides buttons for the timer flags defined in the methodology, i.e., T1 to T7.
- The button naming includes the timer flag names plus a descriptor text which is derived from the analysis of use cases obtained as described in section 10.
- There is an optional text input field "Comments" where the user can enter additional information.
- There is are explicit Submit and Discard buttons. Even when the user hits "Discard", the current data set (timer flag buttons pushed and content of the Comment" field) is recorded/uploaded.

There is button logic aiming at preventing unintended actions as well as providing the required freedom of operation. The following button logic is recommended, but may be chosen differently in different implementations:

- The Submit button acts immediately while the Discard button produces a dialog box asking for confirmation. This is to reduce the number of button taps in the case considered as the most common one.

- The Discard button is permanently enabled. The Submit button is enabled after T3 has been pushed. For handling of time-outs in early stages of the transaction, see the next section.
- The T7 button is permanently enabled. Button T1 enables T2, T2 enables T3. T3 enables T4 to T6. This follows the sequence prescribed by the use case modelling and shall reduce the risk of submitting erroneous data. As T4, T6 and T7 can appear in any order, sequence-forcing after T3 or automatic submission of data cannot be supported.

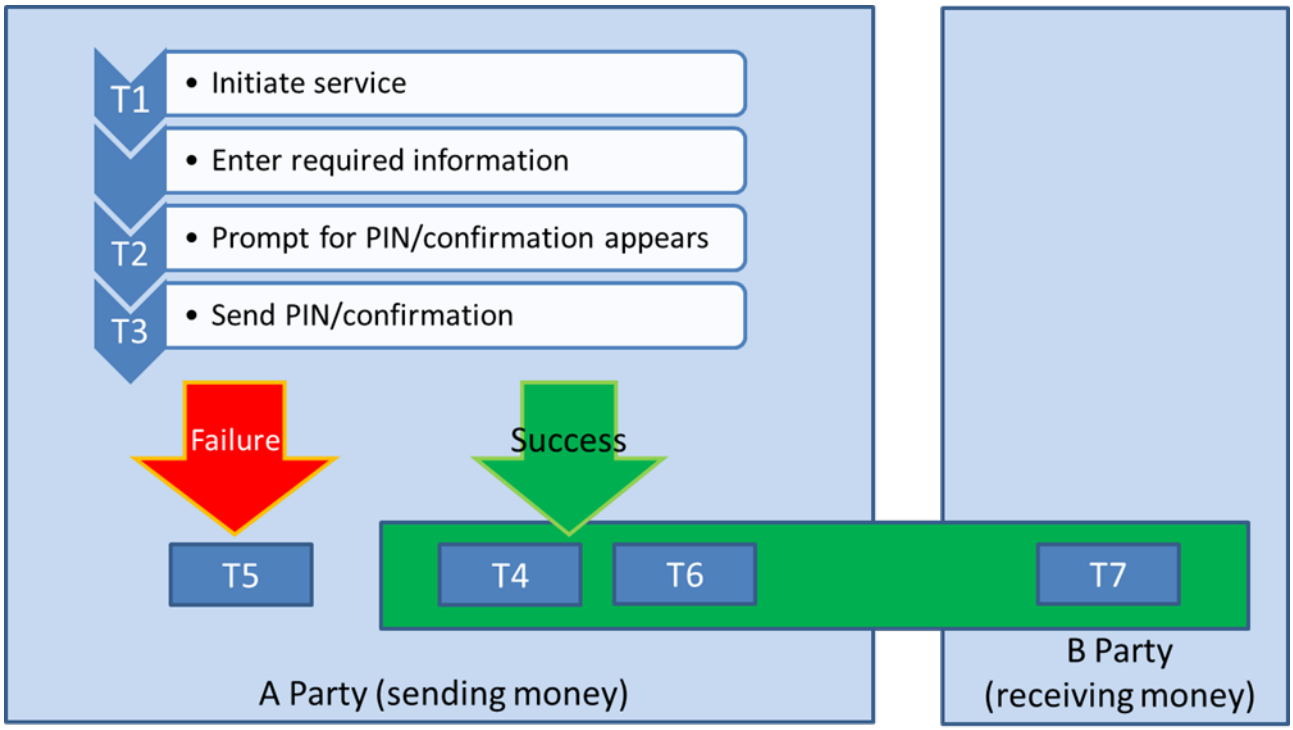
If a specific implementation uses a different set of button logic, adjustments to the practical hints as given in the next section have be required.

Different implementations may make other choices in actual user interface aspects such as screen layout, button-locking logic colorization. It is however assumed that the functional core is the same, so the descriptions in subsequent sections of this methodology refer to the generic functional elements as defined in from above list.

14.2 Practical Application

The following applies for the synchronous testing mode as described in section 13. The Figure below symbolizes the action flow. For asynchronous testing, the B-Party activities are omitted.

Figure 12 – Symbolized flow of action (synchronous testing mode). For the asynchronous testing mode, B-Party activities are omitted.



- A-Party Situations, normal testing workflow
  - Normal flow of actions: Successively tap T1, T2 etc. To conclude and record/upload the data set, tap Submit.
  - No response (timeout) after activating the service by USSD command (T1 has been recorded): Enter "A Timeout" in the Comment field, then tap Discard.
- B-Party Situations, normal testing workflow
  - Getting ready to expect incoming notification: tap T1 (optional). When notification comes, tap T7, then Submit.
  - No notification after agreed time-out period: Enter "B Timeout" in the Comment field, then tap Discard.
- Handling erroneous actions:

- If for some reasons a transaction should be ignored (e.g., because during data entry wrong actions have occurred, or for other reasons), a short text describing the cause should be entered in the Comment field, and Discard being tapped. The wording can either be freely chosen (avoiding texts used to describe A or B-Party situations as defined above), or can be defined to flow a set of terms agreed in the specific campaign. In any case, the text shall be suitable to clearly mark transactions which need to be excluded from post processing.

15 MEASUREMENTS IN THE BACKGROUND

While staying within the general conceptual frame of ITU-T Recommendation P.1502, the context of multi-network or cross-border testing require some differentiation and careful consideration of respective conditions.

For convenience of reading, Table 6 of said Recommendation is repeated here to show in which cases the local mobile network performance has an effect of DFS QoS:

	Well-performing DFS functionality	**Poorly performing DFS functionality**
Well-performing mobile network	High level of overall QoE, only vulnerable to local or temporal impairments of each component	Mobile network performance not relevant/not visible
Poorly performing mobile network	Overall DFS QoE strongly depends on mobile network performance	Low level of overall QoE, no clear dominance of each component

Background testing typically uses a mix of test cases which are considered to be relevant for the DFS-related performance of the mobile network used for the DFS use case, i.e., in the current context, packet-data service performance.
In the case of inter-operator and cross-border operation, based on the general principle that the scenario should be the same for all teams, the following guiding rules apply:

- The network load caused by background-testing should be moderate.
- There should be no country-specific elements; it follows that e.g., web browsing should use standardized reference pages only, or live pages which can reason ably assumed to be general enough.

This leads to the following recommended scenario:

- Fixed-size HTTP download with moderate content size (e.g., 3 Mbyte), and generous time-out/pause values
- Fixed-size HTTP upload with moderate content size (e.g., 1 Mbyte), and generous time-out/pause values
- Web browsing with a rather lightweight standardized reference page, e.g., ETSI Kepler Smartphone

USSD and SMS elements can be added or omitted, based on the assessment of the FTL). USSD can be problematic because a uniform set-up across all teams would need USSD codes which fulfil the requirements of P.1502 for a multitude of countries. In the case of SMS, the set-up of devices would be considerably more complex due to requirements by the Android operating system. If USSD or SMS are not primary for the implementation of the DFS service (also in case of SMS, the use case itself provides information about relative SMS performance), it is recommended to leave them out of the scenario.

16 GENERAL CONSIDERATIONS ABOUT ERRORS IN MEASUREMENTS

Before entering in specific considerations, the usage and definition of the term, 'Error' needs to be to be clarified as this term is used in several conceptual contexts.
'Error' can mean

I) A statistical error in the sense of an error margin. If a quantity is calculated from a limited number of data samples from a system which exhibits, from the user's viewpoint, somewhat random behaviour (such as a failure probability), this quantity will not describe the respective property of the system exactly but only within a given margin. This margin can be calculated based on statistical formulae; respective information can be found in ITU-T Recommendation E.840 or E.804. In short, the only way to reduce this error margin is by increasing the number of samples taken.

II) Errors caused by incorrect reading or transmission of readings, i.e., "human error" in the data collection process. ITU-T Recommendation P.1502 deals extensively with such errors in the context of measurements on DFS. Avoiding such errors requires careful execution of testing and data-collection steps. The check lists and procedures described in the present document are a tool to provide robustness of the measurement process and to reduce the probability of such errors. However, there is always a trade-off between the effort for such checking procedures and impact of actual undetected errors. In general, single errors will decrease the accuracy of measurements. As far as such errors are effectively random in nature, increasing the number of samples is also a means to reduce their impact on output data quality. Applying cross-checks and "logical tests" is also a way to reduce the probability that such errors take place undetected.

   For instance, checking the number of samples against the testing time, by estimating expected sample counts and comparing them to actually received ones, would be a simple first-level way of data quality assurance.
   The way the TAL and DAL lists are constructed, and the usage of field test logs, are also expressions of this strategy. It needs to be mentioned, however, that applying a pre-defined recipe alone is not sufficient. Considering the concrete situation in the field, and applying respective checking steps, are equally important parts of an overall error-reduction strategy.

III) Errors caused by operating errors, i.e., a special type of 'human error' but with an impact on more than one data point. Examples would be insufficient power supply (low-battery condition) which can cause untypical device behaviour; overheating of devices due to insufficient air flow or exposure to heat sources; forgetting to activate functions on the devices etc. The log templates and associated regular checking procedures are designed to provide protection against such errors. Again, these measures need to be complemented by assessment of concrete field situations and respective judgment and definition of additional measures based on actual circumstances.

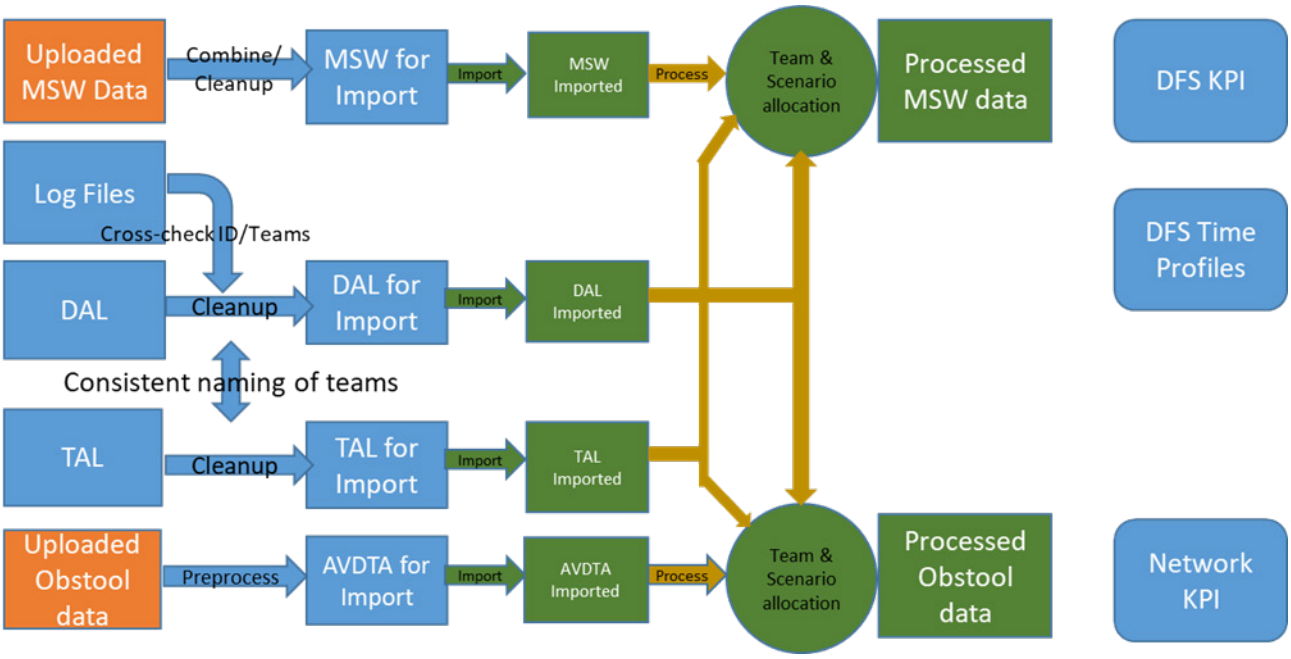
IV) Errors in the implementation of data processing. The way to reduce this risk is running test of algorithms (e.g., SQL queries) with a limited number of data points, and compute reference values manually (typically in a spreadsheet calculation application). Even if pre-defined processing algorithms are provided (e.g., by a set of SQL statements used in previous measurement campaigns) it is advisable to apply such tests, unless it is assured that the processing environment is exactly the same.

17 DATA VALIDATION AND PROCESSING

17.1 Overview

Figure 13 shows a schematic description of the structures and processes for post-processing (see also section 11.2 for a description of the data entities).

Figure 13 – Symbolic overview of data objects and processing



The steps for data cleansing and processing are:

- Complete and validate the TAL; make sure team column headers are consistent with data base requirements (see also 11.6).
- Complete and validate the DAL against field test logs; make sure column headers are consistent with data base requirements (see also 11.7 for reference).
- Cross-validate DAL and TAL, make sure that team names are consistent.
- Cross-validate MSW and ObsTool data, make sure that data source ID's can be resolved to team names and scenario names.

17.2 Plausibility and Validity checks

17.2.1 Basics

The intensity of checks will depend on assumptions about outer conditions of testing; if there are factors which can lead to a higher risk of data loss, checks should be run more often, and vice versa. It is good practice to run these tests on the "data harvest" every day or every other day.

- General yield of data: Check if the number of data items from the MSW and the network performance test roughly corresponds to the overall testing time.
- Cross-checking with GPS data: If the location permits, GPS data can be an important source of information for cross-validation. For instance, the GPS data yield (data points/hour) should correspond to the overall testing time. Also, time information in GPS can provide information for cross-checking device settings. For that purpose, the background-testing tool should provide the original NMEA sentences or an adequate equivalent.
- Cross-checking the session/location logs with the information in the TAL/DAL.

17.2.2 DAL and TAL validity checking

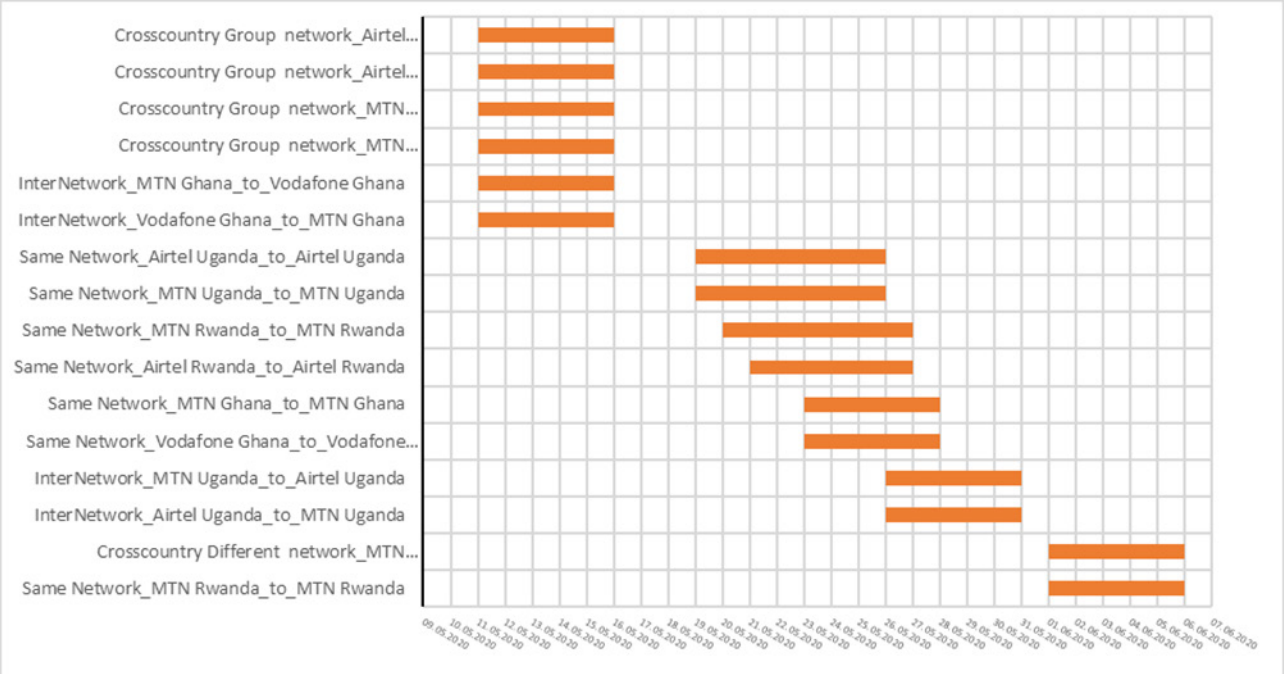
In order to correctly combine the data from different sources (e.g., T1 to T6 from the A-party side with T7 from the B-party side), information about the pairing of teams, and temporal assignment to DFS testing scenarios, is required. This information is provided by the DAL (see also section 11.7) and TAL (see section 11.1). Therefore, DAL and TAL are also imported to the database, and after validity checking, typically processed into respective internal tables with added (constructed) content.

In order to process data, a unique scenario descriptor is required which is used to aggregate (group) data to respective KPI. Depending on the actual TAL structure, such a descriptor can either be included in the TAL directly, or – preferably – be constructed in the data base from basic elements. In any case, validation is required to make sure this indicator is indeed unique, and data can be assigned without creating gaps or duplicates.

It is also helpful to produce unique scenario identifiers as numbers. A scenario name will typically be a rather long string of text which may be impractical to be used in dense tables or graphs. A scenario index, in combination with a look-up table, makes labelling easier.

For TAL validation, it is helpful to create a visualization of the TAL in the form of a GANTT diagram as shown in the example below. With the help of such a visualization, it can easily be checked if all scenario/time ranges are present and consistent. Also, the source table for this visualization can be used to check the scenario names for uniqueness.

Figure 14 – Example of a GANTT visualization of a TAL



17.3 Data Processing

A good practice for data processing is to import data from MSW and network background testing into a central database, and run the final processing there.

In the first step of processing, MSW and network KPI are combined (joined) with respective scenario and team information. In the second step, respective grouping of DFS and network KPI is done based on this information.

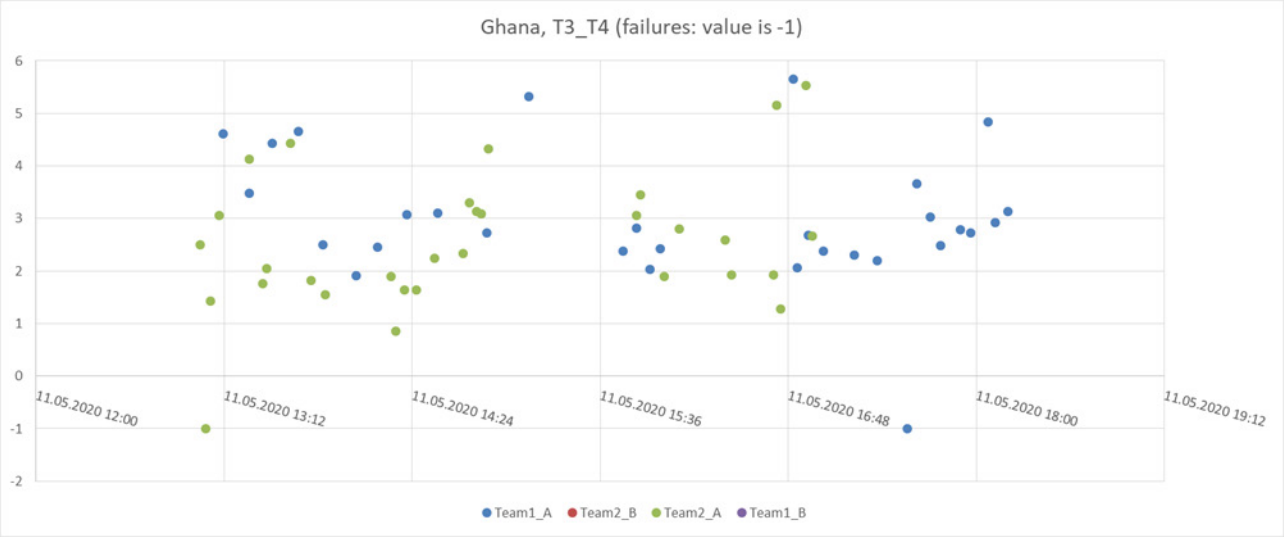
The basis of these join operations is information relating the data items to scenarios. This is done in a multi-step operation. In the first step, technical identifiers are used to connect to the configurations or owner teams. This can be done by creating a look-up table from the DAL with respective join operations, or by assigning these elements directly in respective SQL statements when the processed MSW and network KPI tables are created.

Please note: The exact way of how to run the join operations will also depend on the way the measurements are done. If devices are kept within the same team, and different configurations are used, the device/app ID has to be linked to the Owner Team (see Team Assignment List (TAL) and Device Assignment List (DAL)) and the time range information provided. If devices with fixed configurations are exchanged between teams, linkages have to be made using the device configuration information.
The result of such join operations would be extended MSW and network KPI tables which contain the scenario names used in the TAL, as the element required to do aggregation.

The third and final step of processing is then creating respective table and graphical visualizations. Typically, this is made by either creating tables in a spreadsheet application such as Excel® for conversion to graphics, or directly using graphical front-end tools.

17.4 Time Profile

A time profile is both useful for data validation and for reporting.
The time is an x-y diagram created from MSW. It is basically a scatter plot where, on the x axis, date and time is shown while on the y axis, a selected indicator is shown.
The following example shows a time profile for a pair of teams.



From such a graph, various information is directly visible:

- The time range of tests.
- The value range (band).
- These tests were done in asynchronous mode, i.e., there are no B-Party events.
- There are a few failure cases (data points having the value -1); it would have to be clarified if these were real or false positive cases.

18 BACKGROUND TESTING OF MOBILE NETWORKS

18.1 General considerations

The considerations and recommendations on mobile network background testing provided in P.1502 fully apply as running such tests on the local network do not depend on the test scenario. The following is therefore meant to be as an extra angle of view on the matter.

As outlined in P.1502, the effect of mobile network performance on overall DFS QoS depends on the performance and interplay between DFS infrastructure and the network. Only if the DFS infrastructure works very well, i.e., the processing times are consistently short, mobile network performance may become the defining or limiting factor in overall DFS QoS. With slow or strongly fluctuating DFS performance, the influence of the mobile network may not be visible in output data at all.

Also, if mobile network performance is a matter of interest at all depends on the overall scope and goals of a campaign. Therefore, the effort made when testing mobile network performance (e.g., if devices are allocated for testing, or the budgeted cost of mobile data plans) is typically decided case by case. For instance, if tests are done stationary, spot testing may be sufficient to assess the mobile network coverage quality, instead of running data-intensive testing all the time.

18.2 Testing Tools

Generally, all applicable network performance testing tools can be used. For practical reasons, tools which come as an Android app running on "out of the box" mobile phones may be preferable from a cost point of view.

18.3 KPI

The following set of use cases and KPI provides good overview at reasonable effort. It does not include SMS or USSD for the following reasons:

- SMS is not a primary transport service for DFS related information. It is only used to transfer notification SMS to the B party (which is irrelevant in asynchronous mode anyway), and for the A side SMS acts only as a secondary indicator.
- As can be seen in respective reports (see References), USSD performance has shown not to be highly correlated with DFS performance. Also, in a multi-network, multi-country campaign, it will be hard to find USSD codes which work for all involved networks. In summary, the effort for including USSD to such campaigns should be carefully considered form a cost to value point of view.

The definition of valid TA excludes transactions which are taken via Wi-Fi, were interrupted by the user ("user break") or are masked-out otherwise. Also, through joining with the TAL, there is an effective time-windowing to exclude TA taken outside the date range of respective scenario. Due to the fact that measurements were taken stationary (in the same location), there is, however, no time windowing with respect to MSW time ranges.

MDR values are, different from standard MDR averaging, taken over all TA including unsuccessful ones. This avoids biasing towards higher expected values which occurs when timed-out transactions are excluded from averaging.

ST values are calculated over values from successful TA only to avoid inconsistencies by clipping. When interpreting data, success rates need to be considered along with ST values.

When setting up a scenario for network testing, it also needs to be considered where respective content is hosted. The effort to be taken is, again, a matter of scope and purpose of measurement. If network KPI shall only have an indicative or secondary character, a simple approach i.e., by hosting all content on the same server can be taken (located in one of the participating countries, or elsewhere). If precision measurements are intended, multiple server locations with high supported bandwidth may be required, possibly accompanied by calibration and validation testing.

Test case	KPI
Web Browsing (ETSI Kepler SP reference web site)	**End to end session time (ST_E2E)** in case of successful transactions. In contrast to the Session Time defined in ETSI TS 102 250-2/ITU-T Rec. E.804, the time window begins with the start of web site download (not with reception of the first package) **End to End Success Rate** (percentage of transactions successfully completed, from all valid TA. A valid TA is a TA run via mobile network (not via Wi-Fi), and not blanked out by e.g., a User Break indication.
HTTP DL with 3 Mbyte file; time-out 30 sec	**End to end session time (ST_E2E)** in case of successful transactions. Analogously to Web browsing, this ST includes the initial start time. **End to End Success Rate** (percentage of transactions successfully completed, from all valid TA. Evaluation is done in fixed-size mode, i.e., a TA which ran into a time-out is not counted as successful. **Mean Data Rate End To End (MDR_E2E)**: Effective data rate. This value is also output if the result if the TA is unsuccessful (e.g., dropped or ran into time-out); in that case the transferred data up to the stopping point, and the time expired, is used to compute the MDR.
HTTP UL with 1 Mbyte file in fixed-time mode; time window 30 sec (hybrid mode)	In hybrid mode, the TA ends either when the intended data volume is transferred, or the time window is expired. In this mode, reaching the end of the time-window does not result in the result "unsuccessful". If desired, a computational "unsuccessful" state can be created by evaluating the TA duration. **End to end session time (ST_E2E)**: By computation, this value is created only when the end of the time window is not reached (to stay consistent with standardized KPI computation. Analogously to Web browsing, this ST includes the initial start time. **End to End Success Rate** (percentage of transactions successfully completed, from all valid TA. Evaluation is done in "computational fixed-size mode", i.e., a TA which ran into a time-out is not counted as successful. **Mean Data Rate End To End (MDR_E2E)**: Effective data rate. This value is also output if the result if the TA is unsuccessful (e.g., dropped or ran into time-out); in that case the transferred data up to the stopping point, and the time expired, is used to compute the MDR.

ANNEX 1. FULL LOCATION LOG SHEET

Location Log Sheet

Team ID		Sheet started (hh:mm)	
Team Leader		Sheet completed (hh:mm)	
Location		Sheet saved/photographed	
Date		Sheet Photo submitted	

Scenario	

Initial check of conditions and measurement set-up at location			
	DFS Device	Observer Device	
Device Type			
Device ID (IMEI)			
FIMSW Version	n.a.		
FIMSW ID	n.a.		
DFS Observer Version	n.a.		DFS Observer started/running?
Mobile Network			(pls. check) ☐
RF Level (bars)			
Battery Level (bars)			
Charger (y/n)			
Date/time correct			

Regular checks (every ca. 2 h)			
	DFS Device	Observer Device	
Mobile Network			Time of check:
RF Level (bars)			
Battery Level (bars)			
Charger (y/n)			
Date/time current?			
Mobile Network			Time of check:
RF Level (bars)			
Battery Level (bars)			
Charger (y/n)			
Date/time current?			
Mobile Network			Time of check:
RF Level (bars)			
Battery Level (bars)			
Charger (y/n)			
Date/time current?			

Final, before leaving the location			
	DFS Device	Observer Device	DFS Observer Data Upload
Mobile Network			Measurement stopped ☐
RF Level (bars)			Data upload started ☐
Battery Level (bars)			Data upload completed ☐
Charger (y/n)			
Date/time current?			

ANNEX 2. SHORT FORM LOCATION LOG SHEET

Short Form Location/Session Log Sheet

Team ID		Sheet started (hh:mm)	
Team Leader		Sheet completed (hh:mm)	
Location		Sheet saved/photographed	
Date		Sheet Photo submitted	

Scenario	

Initial check of conditions and measurement set-up at beginning of the testing session

	DFS Device	Observer Device	
Device Type			
Device ID (IMEI)			
FIMSW Version	n.a.		
FIMSW ID	n.a.		
DFS Observer Version	n.a.		DFS Observer started/running?
Mobile Network			(pls. check) ☐
RF Level (bars)			
Battery Level (bars)			
Charger (y/n)			
Date/time correct			

Regular checks (every ca. 2 h)

	DFS Device	Observer Device
Time	Network/Power conditions okay?	Network/Power conditions okay?
	☐	☐
	☐	☐
	☐	☐

Final, after ending the testing session

	DFS Device	Observer Device	DFS Observer Data Upload
Mobile Network			Measurement stopped
RF Level (bars)			Data upload started ☐
Battery Level (bars)			Data upload completed ☐
Charger (y/n)			
Date/time current?			

Endnotes

[1] All cited material is listed with full name and, where applicable, with a link for downloading in the References section.

[2] P.1502 is assuming that both the A and B party device are in the same place. Of course, even for intra-country, intra-network testing it is possible that operation in different places is desired. Therefore, the present document provides an extension to P.1502 even in this basic case.

[3] Data processing and cleansing can compensate for a certain amount of simplification or omissions in the data collection process, e.g., by inferring missing or incorrect information from circumstantial data. However, this cannot be taken for granted, so careful data collection in the field is essential for a high yield of useful data, and high quality of resulting information.

[4] This also provides the appropriate degree of genericity, as there is no specific application prescribed for these tests.

International Telecommunication Union
Place des Nations
CH-1211 Geneva 20
Switzerland