CYBERNETICA

ESTAT 2019.0232 Solution Architecture

Hendrik Eerikson, Armin Daniel Kisand, Baldur Kubo, Angela Sahk, Ville Sokk, Riivo Talviste, Toivo Vajakas

Technical document

Version 1.4

November 24, 2021

47 pages

Doc. Y-1440-2

Disclaimer. This document was prepared by Cybernetica AS as part of a procured project under Service Contract No ESTAT 2019.0232 (Ref. Ares(2020)2309804 - 30/04/2020). The opinions expressed in this document are those of the authors. They do not purport to reflect the opinions, views or official positions of the European Commission or its members.

Contents

1 Introduction

1.1 Overview

The ESTAT 2019.0232 project is a collaboration between Cybernetica and Eurostat to develop a proof-of-concept solution for passive mobile positioning data analysis which preserves the privacy rights of the data subjects. State-of-the-art trusted execution environment technology and pseudonymisation are applied in the solution to ensure privacy by design in calculations and output.

The project scenario involves a mobile network operator (MNO) and national statistics institute (NSI). The NSI uses the project solution to compute longitudinal analysis over a longer period by combining the location data of MNO subscribers with confidential auxilary data from the NSI.

The project approach honours the principles of data minimisation, privacy-by-design and purpose specification that lie at the foundation of the General Data Protection Regulation ("GDPR") of the European Union.

1.2 Document Scope

This document provides an architectural overview of the Solution. It is intended to capture and convey the most significant architectural decisions that have been made on the system.

This document is intended to be read in conjunction with the Solution Analysis document.

The full list of delivery documents:

- Solution Analysis
- Solution Architecture
- DPIA Evaluation Report
- User Guide for NSI
- User Guide for MNO-VAD
- User Guide for Auditors
- User Guide for MNO-ND
- Sharemind HI Documentation
- Sharemind HI ToS
- Sharemind HI License
- Synthetic Test Data Generation

1.2.1 Expected Audience

The expected audience of this document are:

- IT support admins on client premises;
- Technological stakeholders in involved organisations (NSIs, MNOs, Eurostat);
- Cybernetica's devops team.

1.3 Glossary

DWH: Data Warehouse. In this document refers specifically to mobile positioning database in MNO-VAD department.

ETL: Extract, Transform, Load – three database functions that are combined into one tool to pull data out of one database and place it into another database (www.webopedia.com).

IMSI: An international mobile subscriber identity (IMSI) is a unique number associated with mobile subscribers.

MNO: Mobile Network Operator.

MNO-ND: Network Department in Mobile Network Operator organisation.

MNO-VAD: Value Added Services Department in Mobile Network Operator organisation.

PoC: Proof of concept.

SDC: Statistical Disclosure Control.

SGX: Intel Software Guard Extensions (SGX) is a set of security-related instruction codes that are built into some modern Intel central processing units (CPUs).

Subscriber: The user of the services of the MNO; owner of the positioned device.

TEE: Trusted Execution Environment.

2 Architectural Requirements and Considerations

2.1 Functionality

Solution functionalities originate from

a) functional requirements of business cases; and
b) from the use of a specific privacy-enhancing technology platform: Sharemind HI.

2.1.1 Functional Requirements: Business Cases

A general description of the selected business cases is provided in the deliverable “Eurostat project business analysis document”. Within this document, we only re-iterate functional requirements of the identified business cases that are most important from the architectural point of view.

2.1.1.1 Periodic Pseudonymisation of Mobile Positioning Data

#	Requirement description
B0-1	The Solution must generate periodic pseudonymisation keys. The key generation shall be performed inside a TEE.
B0-2	The Solution must load mobile positioning data from MNO-ND.
B0-3	The Solution must perform pseudonymisation of mobile positioning data, using the periodic pseudonymisation key.
B0-4	The Solution must output pseudonymised mobile positioning data to be consumed by MNO-VAD.

2.1.1.2 Statistical Analysis of Periodic Human Mobility Footprint Data

#	Requirement description
B1-1	The Solution must allow an authorised user to import pseudonymised human mobility footprint data, based on passive mobile positioning, into the Sharemind HI server.
B1-2	The Solution must allow an authorised user to import calibration data into the Sharemind HI server to be used together with pseudonymised mobile positioning data for the computation of final aggregate statistics.
B1-3	The Solution must allow an authorised user to start an authorised task.

#	Requirement description
B1-4	The Solution must reverse mobile positioning data pseudonymisation, inside TEE, to enable longitudinal analysis of data.
B1-5	The Solution must compute, inside TEE, aggregate values from the mobile positioning data. Computations are detailed in Section 3.3.1.
B1-6	The Solution must allow an authorised user to download an encrypted copy of the computation result and to decrypt it.

2.1.2 Functional Requirements: Sharemind HI Platform

Here we list functional requirements that relate to the security guarantees provided by the chosen TEE platform. These are not directly connected to the computation, i.e. business case of the Solution.

#	Requirement description
P-1	The Solution must enable users of the Solution to validate the integrity of the deployment (remote attestation).
P-2	The Solution must enable authorised users to validate the correctness of *Dataflow Configuration*.
P-3	The Solution must enable authorised users to audit the computation results and logs.
P-4	The Solution shall prevent starting the statistical analysis application unless all the Enforcers have signed the respective *Dataflow Configuration*.

2.2 Supplementary Requirements

The following sections cover the below supplementary requirements:

- Compatibility;
- Security;
- Performance efficiency;
- Limitations of architecture.

2.2.1 Compatibility

2.2.1.1 Co-existence

1. The Solution shall operate on Linux (Ubuntu 20.04).
2. The Solution shall work on CPUs with Intel SGX support enabled.
3. The Intel SGX support of the Solution shall be implemented with and will use the Intel SGX platform toolchain[1]:
   a) *Intel SGX SDK* (BSD license),
   b) *Intel SGX Platform Software (PSW)* (BSD license),
   c) *Intel SGX Driver* (3-clause BSD license).
4. The Solution shall, to the extent feasible, preserve the current mobile positioning data pseudonymisation process (i.e. pseudonym format and/or length) at the MNO.

[1] Intel SGX platform software: `https://01.org/intel-softwareguard-extensions`

5. The Solution shall allow the MNO-ND to use custom locally generated periodic pseudonymisation keys (i.e. not ask it from the *pseudonymisation_key_enclave* service). This is required to increase robustness, for example, when the pseudonymisation functionality or the Solution is unavailable. In this case the MNO is still able to carry out its internal intraperiod data analysis based on short-term pseudonyms. Note that it is impossible to reverse MNO-ND generated periodic pseudonymisation keys within the Solution.

2.2.1.2 Interoperability

1. The Solution shall communicate with the Intel Attestation Service (via the Sharemind HI platform). On communication protocol version change (e.g. due to a patched vulnerability), Cybernetica AS will notify the Solution host.

2.2.1.3 Availability and Fault Tolerance

1. The Solution shall be a fail-fast system due to security considerations, i.e. it shall stop operation rather than attempt to continue a possibly flawed process.
2. The Solution shall handle errors and exceptions gracefully before stopping the flawed operation or process, avoiding system crashes and other unintended behaviour.
3. The Solution shall continue normal operation after a gracefully handled failure if and only if this does not negatively impact further functioning of the Solution.
4. The Solution shall withstand a graceful failure of the analytics tasks and continue normal operation.

2.2.2 Security

2.2.2.1 Confidentiality

1. The Solution shall provide technical means and work processes to prove security of the configuration of the underlying TEE platform.
2. The communication between the client application and TEE must be encrypted.
3. The Solution shall not reveal uploaded data to any other user besides the one who uploaded it. This applies to the pseudonymised mobile positioning data by the MNO-VAD and calibration data by the NSI.
4. The Solution shall not reveal re-identified (with periodic pseudonyms removed) mobile positioning data to any user, it is usable only by the code inside the TEE of the Solution.

2.2.2.2 Integrity

1. The Solution shall provide technical means and a procedure to attest integrity of the TEE platform in use.
2. The Solution shall provide technical means and a procedure to attest integrity of the analysis code inside the TEE of the Solution.
3. The Solution shall protect integrity of persistent data stored by the TEE of the Solution.

2.2.2.3 Authenticity

1. The Solution shall provide technical means and a procedure to attest authenticity of the TEE platform in use.

2. The Solution shall provide technical means and a procedure to attest authenticity of the analysis code inside the TEE of the Solution.

2.2.2.4 Non-repudiation

1. The Solution shall keep a cryptographic log of all client interactions.

2.2.2.5 Accountability

1. The Solution shall provide means to detect and audit any modifications to the code inside the TEE of the Solution.
2. A work process must be specified by the Solution host (or deployment managers) in order to handle new discovered weaknesses in either the selected TEE or Sharemind HI. These will be reported by the Sharemind HI product team to licensed users.

2.2.3 Performance Requirements

Performance estimates depend on the algorithm and on the characteristics of the input data. To understand performance estimation details one should be familiar with the algorithm description in the Annex 7.1. At the time of writing this document some characteristics of the real input data are not fully known and the estimates can change.

2.2.3.1 Assumptions for Performance Criteria

Note: Tile size, in input data S and in output reports, is assumed to be circa 1 $\times$ 1 km. Tile size affects the count of tiles. Tile count affects the performance and required resources.

The following assumptions have been made in relation to the proof-of-concept Solution performance:

1. Number of unique subscribers (IMSIs) during report period: 50 000 000
2. Report period duration: 180 days
3. Period of footprint data (H) input: 24h
4. Unique tiles count over report period: 1 000 000
5. Number of Reference Areas: 200
6. Average footprint tiles count per mobile subscriber in single H data: 10
7. Average unique footprint tiles count per mobile subscriber over report period (in S data): 200
8. Average number of “prevalent” visit tiles (Y) for IMSI: 8

2.2.3.2 Time Behaviour

1. Maximum time of daily S aggregation calculations in “calculate incrementally” scenario: 4 hours
2. Maximum time of final report calculations in “calculate incrementally” scenario: 4 hours
3. Maximum time of all report calculations in “calculate in batch” scenario: 7 days

2.2.3.3 Resource Utilisation

1. Maximum RAM needed in Sharemind HI platform instance: 16GB
2. Maximum disk space needed internally for HI calculations: 1TB

3. Maximum persistent disk space for storing single day H file in HI-compatible format: 0.1TB
4. Maximum persistent disk space for NSI inputs and reports: 0.1TB

2.2.4 Maintainability and Portability

2.2.4.1 Modularity and Reusability

1. The Solution shall, to the extent feasible, be built in modular fashion so individual modules could be replaced. For example, replacing the pseudonymised data storage interface that might differ between MNOs. Report calculation algorithm updates should be possible without major changes to rest of the Solution.
2. Versioning of software shall be implemented to govern module lifecycles (subject to limitations described in Section 2.2.6).

2.2.4.2 Analysability

1. The packaging and deployment processes of the Solution shall be deterministic and sufficiently documented for proceeding code audits.

2.2.4.3 Adaptability

1. The Solution shall utilise modular design to allow relatively easy changes to statistical report calculation algorithms and integration with the MNO ETL processes.

2.2.5 Limitations of the Architecture

Upholding security and privacy guarantees is of utmost importance in Sharemind HI development. Sharemind HI relies on the Trusted Execution Environment (TEE) technology to provide the security guarantees. A TEE isolates the security sensitive parts of an application from the rest of the system with the help of trusted hardware. The TEE technology used in Sharemind HI to implement the privacy-preserving data processing is the Intel® Software Guard Extensions (SGX) which is available in modern Intel® processors.

Main design constraints associated with Sharemind HI are:

- The security model of Sharemind HI relies on the security guarantees of Intel SGX. Trust in the Solution will be rooted in Genuine Intel SGX platform TEE and Intel SGX Attestation Service (IAS).
- Sharemind HI *Task enclaves* are bound by the programming languages and their allowed functionality that are supported by the Intel SGX SDK:
  - Supported programming languages include C99/C++11 with some instructions removed[2].
  - An SGX enclave can call only functions that are statically linked to the enclave image.
  - An SGX enclave runs only in *user mode* (ring 3).
  - An SGX enclave can communicate with other processes only via *ecall* and *ocall* mechanisms.
- Current role setup adds constraints to processes, build, configuration of any application built on top of Sharemind HI;

[2] Intel® Software Guard Extensions Programming Reference, 3.6.1 Illegal Instructions. https://download.01.org/intel-sgx/sgx-linux/2.13/docs/Intel_SGX_Developer_Guide.pdf

- Each deployment of Sharemind HI has to align to the platform and application lifecycles;
- For Sharemind HI guarantees to apply, application development needs to be security and privacy oriented, which may ultimately require more effort and resources.

2.2.6 Limitations of the Proof of Concept

2.2.6.1 Upgrading Task Enclaves

In Intel SGX, cryptographic secrets (e.g. keys used to encrypt data written to disk) are tied to a specific version of an enclave. Any modifications to the enclave code yield a new version and invalidate access to the previously used secrets for the new enclave. This feature is designed to protect against a malicious host replacing the audited and privacy-preserving code of an enclave with a malevolent one to steal secrets meant only to be visible for that particular enclave.

In the context of the Solution it means that any modification to either of the two task enclaves (*pseudonymisation_key_enclave* or *analytics_enclave*) invalidates access to persistent data (see 3.3.6) as well as all user-uploaded inputs. Therefore, with each task enclave code update, mobile positioning data that is to be used in any future longitudinal analysis (e.g. data for the ongoing reporting period) has to be pseudonymised again by the MNO-ND, uploaded to the Sharemind HI server by the MNO-VAD and re-analysed by the corresponding task enclave. The same holds true for any user input (e.g. calibration data by the NSI).

The above is only a limitation of the proof of concept. For the long-term deployment of the Solution it is possible to solve this limitation by implementing a secure export-import mechanism for the cryptographic secrets. Additionally, this would also address the threat D_ds4 in the DPIA Evaluation Report. Such functionality can be made available either explicitly for the current Solution or handling it implicitly inside the Sharemind HI platform itself.

2.2.6.2 Threats at Data Import Process

The DPIA Evaluation Report has identified several data tampering threats (see privacy risks R4 and R8 and threat T_p2 in the DPIA Evaluation Report) and a possible data leakage scenario (see threat I_p2) in the mobile location data import process at the MNO-VAD. In a future version of the Solution, we propose to add additional controls as stated in risk R8: encrypt the whole mobile location record at the pseudonymisation component instead of just the subscriber identifier and move the data pre-processing step of the MNO-VAD also to the TEE part of the Solution. In this case, no-one can read or modify any attributes in the mobile location records as they move from the MNO-ND to the MNO-VAD. Technically, it is possible as the pseudonymisation used in the Solution is in fact encryption with a key that is only available in the enclave.

Unfortunately, this feature would break the Solution's compatibility with the current data pseudonymisation and short-term analysis process implemented at the MNO-VAD. If this is not wanted, there are less invasive enhancements to the Solution that mitigate some of the more specific risks:

- Avoid temporary files in the data import process and if possible, send pseudonymised location data directly from the MNO-VAD DWH to the Solution (see threat I_p2 in the DPIA Evaluation Report).
- Remove country information from the pseudonymised mobile location records or add country information (e.g., total number of different countries) to the analytics enclave log. The

latter discourages specific filtering attacks at the data import process (see privacy risk R4 and threat T_p2 in the DPIA Evaluation Report).

3 Design

3.1 Overview of the Solution Architecture

The following organisations are involved in a full-fledged deployment of the Solution:

- NSI
- MNO
- Cybernetica AS

The NSI deploys the client software to optionally import census data files, run the report functionality and gain access to analysis results on the Sharemind HI server.

The MNO deploys the Sharemind HI server and also necessary ETL integrations in order to provide input data for statistical calculations. Within the MNO two different departments (zones) are distinguished – MNO-ND and MNO-VAD. Within the the MNO-VAD perimeter is the Sharemind HI zone that has improved security guarantees, based on careful use of Intel SGX TEE functionality. Figure 1 describes overall component level architecture of the Solution.

Cybernetica AS delivers the installable software for both the NSI and the MNO, with documentation. For testing the PoC, Cybernetica also provides proxy for the Intel Attestation Service.

3.1.1 Calculations Workflow for Statistical Report

Reports created by the Solution contain statistics derived from the fusion of MNO and NSI input data (NSI input is optional). The specific functions are detailed in Section 3.3.1 and report data structures in Section 4.3. Figure 2 illustrates the workflow for the statistical calculations use case.

The following steps are taken in the report producing process:

1. The system has been started: The MNO-VAD periodically creates H files and periodically runs a script that runs the analytics task (enclave) for each prepared H file. If no active report request is present then the analytics task does nothing during the call.
2. The NSI initiates the report calculation by providing the necessary inputs and a request for a set of reports for a certain time period. Multiple requests can be uploaded; they will be processed in order of upload. The inputs provided by the NSI will *not* be visible to the MNO-ND nor the MNO-VAD. Sample input: census data raster file. Sample request: "*calculate FUF and D' statistic reports for the 1st quarter of year 2021*" For the PoC, the NSI request is expected to cover non-overlapping and ordered dates in the future.
3. When the HI analytics enclave is run by the MNO-VAD then the analytics enclave checks by polling the topic "NSI_input" whether the request has arrived and writes into the "application_log" topic the corresponding status message. The behavior is described below

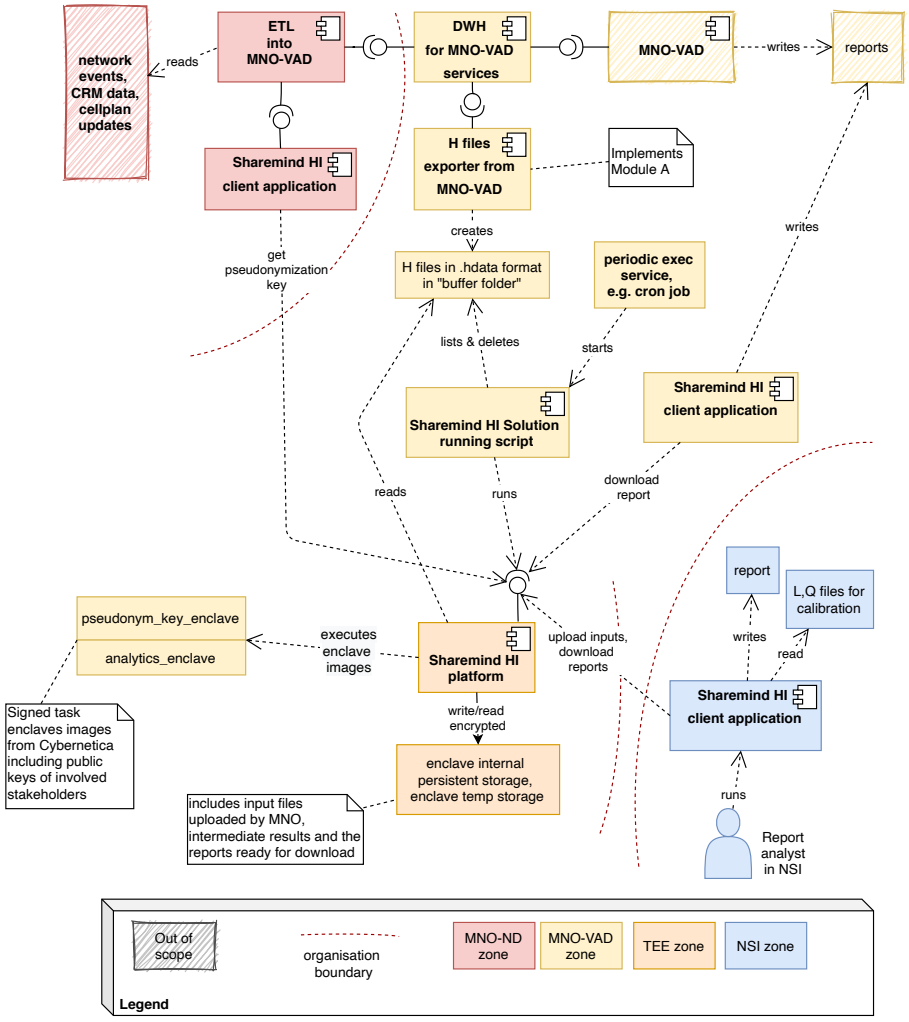

Figure 1. Communication between the components of the system. Colour indicates different zones. Arrows point from proactive (client) component to reactive (server) component.

as a state machine of analytics enclave processing, see Figure 3. Both the NSI and the MNO-VAD can check the start of calculations by reading the topic “application_log” and inspecting the dataflow configuration.

4. The Solution will listen to exported H data and aggregate it into intermediate result (S in algorithm description). This step will repeat until the Solution (either the enclave or the application to invoke `taskRun`) detects that intermediate data is complete and the reports can be produced.
5. The Solution calculates from the intermediate aggregations the final result reports as specified in the report request in step #2.

6. Status is updated and notifications[3] are sent to both the NSI and the MNO-VAD that the reports are available for download.
7. If more requests are in the queue then return to step #3.

All produced reports stay available indefinitely. A delete process is not specified in this PoC.

[3] Sharemind HI supports directly only polling of status . Push notifications can be integrated on the client side. We provide sample scripts which are meant to be run periodically through cron that poll the state and transform updates into a “push” by calling a “callback” script given as an argument. The callback script is then given the status as arguments itself, like “request X at n %” which the respective admin can then integrate into their reporting system, like emails. For the NSI, we provide an invocation for purely this task. For the MNO-VAD, they already have a script that is meant to be run by cron, which automatically processes the H files and controls the analytics enclave state, and this script invocation will also have the described callback capability.

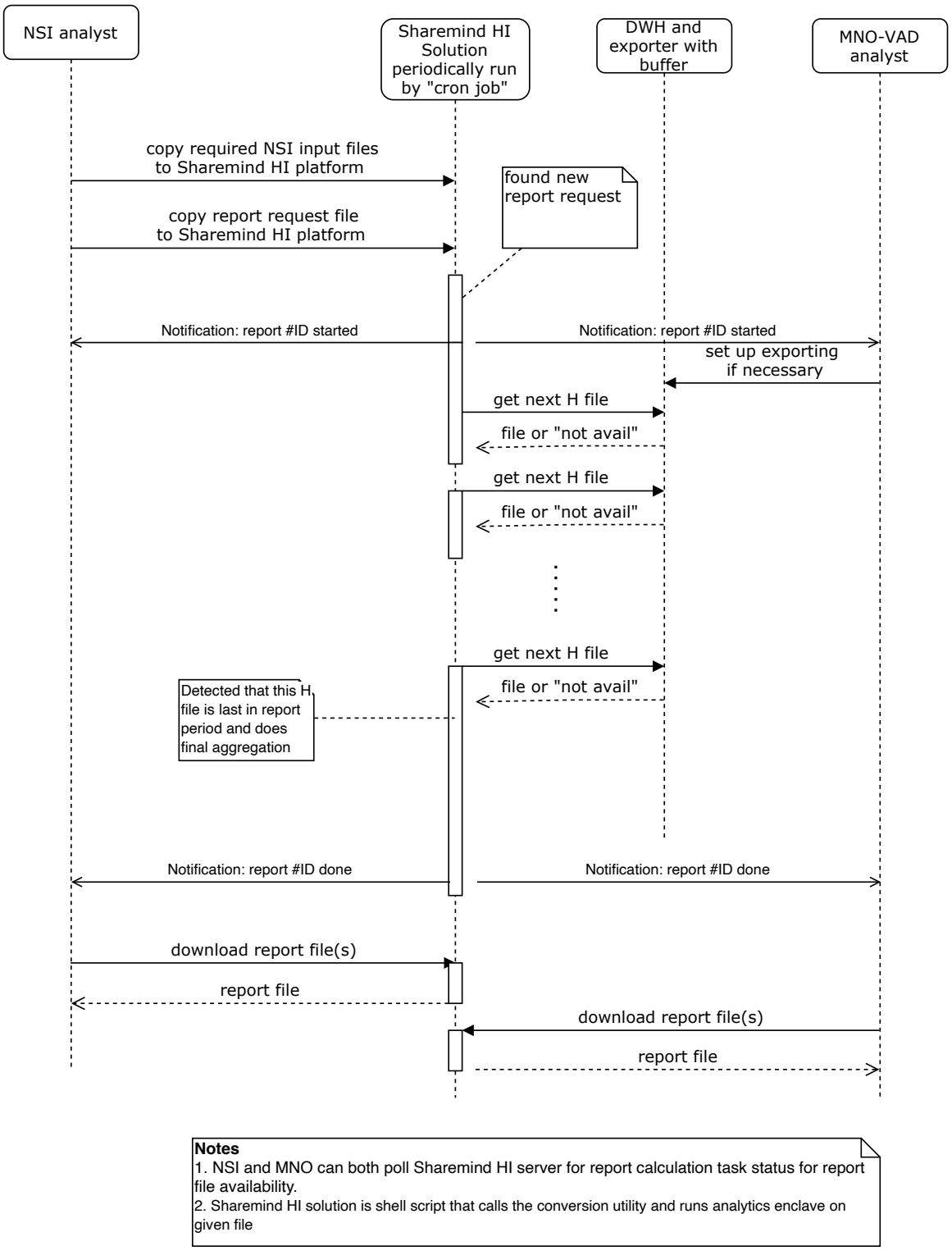

Figure 2. Sequence of activities during statistical report calculations.

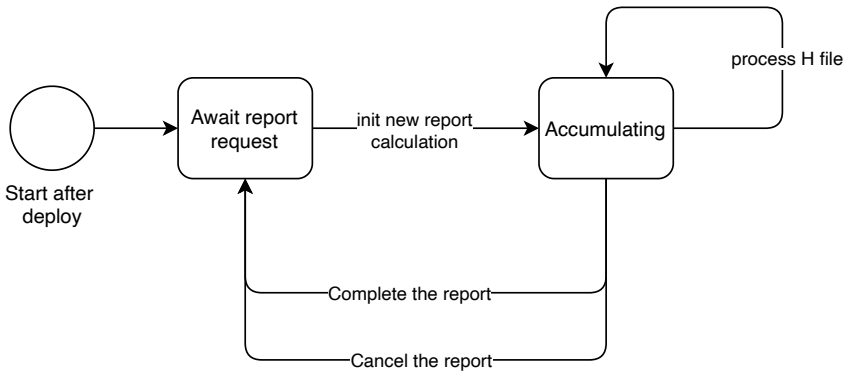

Figure 3. Simplified state machine of analytics enclave processing.

3.1.2 NSI Report Request and Download

The NSI uses a wrapper script to talk to the Sharemind HI server. There are two relevant invocations: One for uploading a request and input data, and one for downloading the results. Uploading a request will be executed manually by an operator and takes the following arguments:

- First period (date),
- Last period (date),
- Path to the reference areas csv file as described later,
- Path to the census residents csv file as described later,
- Path to the HI client yaml conf file

Downloading and sending a status request is done through another invocation of the script which is meant to be executed periodically by a cron job. It receives the following parameters:

- Working directory: where to store information about previously sent progress reports to prevent duplicate reports.
- Output directory: where to place the downloaded reports.
- Another program to call to provide a progress report.
- Path to the HI client yaml conf file

3.1.3 Communication Between H Files Exporter and Sharemind HI Solution.

3.1.3.1 H Files Exporter

The H file exporter makes queries to the positioning data from the MNO-VAD DWH and aggregates the results into H files by means of Module A. Integration between the DWH and the H files exporter is outside the scope of this document. The resulting H files are written into a designated folder `Hfolder` where the periodic task of the Sharemind HI solution is looking for necessary files and sending them into the Analytics Task in the Sharemind HI Server for processing.

Pseudocode:

```
while (wallclock < end_of_processing_agreement) {
  // avoid disk space overflow
  if (H_file_count_in_Hfolder > max_H_buffer_size) {
    delete_oldest_Hfile();
```

```
  }
  if (data_for_next_H_available) {
    run_dwh_query_with_ModuleA();
    save_hdata_file;
  }
}
```

Note: the binary H data file format was chosen to minimise integration problems for Sharemind HI. Module A can produce it directly in code or via the CSV to binary data file converter which is a command line utility provided by Cybernetica AS. File formats are described below in chapter 4 “Data description”.

3.1.3.2 Sharemind HI Periodic Task

The process business context of the task is as follows: The NSI and the VAD have a contract that the VAD provides data for some continuous date range in the future. The NSI may then divide this range into subranges to their liking and sends either a single NSI request or a series of NSI requests with strictly increasing dates to the HI server. It is the VAD’s responsibility to atomically place H data files with a specific filename pattern into the import directory. The files shall be created transactionally, in monotonically increasing time order. It is the responsibility of the Solution to give the files to the analytics enclave and remove the processed files. Therefore, we provide a script to be executed by a cron job which is tailored for this process.

The overall idea is as in the following pseudocode, with more explanation afterwards:

```
lock some lockfile to prevent parallel script execution (and unlock on exit)

while(true) {
    if (some instance is running or pending) {
        exit 0 // Let it finish.
    }

    if (not processing a report) {
        schedule and wait for looking for a new NSI report
        continue if new report was found, exit otherwise
    }

    if (last instance is a failed report generation) {
        // Either a manual report generation,
    // or after the last H file was supplied
        schedule and wait for cancelling of report
        continue
    }

    // At this point we know that there is an active report request for some
    // date range.

    date_range = determine missing H file dates // from DFC and date range
```

```
    // assert: date_range is not empty, as this means the last one was a failed
    // report generation, and this case was handled earlier.

    for (date : sorted(date_range)) {
        if (H file for date exists) {
            append ‘stat H_file‘ to metadata file // preserve existing meta data
            schedule and wait for processing of H file for date
            delete processed H file
        }
    }

    if (no newer H files exist) {
        // There could appear more H files for the current report request
        exit 0
    }

    if (current report request is not finished) {
        // At this point, all available files are for dates after the current
        // NSI request, so it must be finished to activate the next NSI request
        // which is by business-definition in the future and may process these
        // waiting files.
        if (some H files have been processed in the current NSI request) {
            schedule and wait for finishing of report
        } else {
            schedule and wait for cancelling of report
        }
    }
}
```

The script handles the conversion between dates provided by the H file names and periods which can be understood by the analytics enclave.

State management: The script queries the state from the HI server. With the help of publicly visible arguments and selected download of application logs, the script can deduce:

- Whether the enclave waits for an NSI request (application log)
- What are the bounds for the current NSI request (application log, this information repeated on each H processing invocation)
- Which task instances finished an NSI request (has an output `fingerprint_report`)
- Which task instances cancelled an NSI request (has a single argument `cancel`)
- Which task instances are part of the current NSI request processing (have exactly two arguments `period` and `file`, and are after the last task instance which finished or cancelled an NSI request, if there is one)
- Which dates have been processed already (task instances that is part of the current NSI request processing and has status `Finished`)
- Which dates failed to be processed (as previous but status `Failed`)
- Which dates are currently being worked on (as previous but status `Pending`)

For the state deduction, the script ignores incorrectly invoked task instances, which contain a

special error code in their error message `:AE01:`.

The script looks in an import directory for H files with a specific pattern. Each data file shall be accompanied with a metadata file. The metadata file contains human-readable info about file processing, to be included into application-data logging. The data file and metadata file name and format are specified below in chapter 4 “Data description”.

In addition, the script compiles a progress report which will be given to a callback script provided as an argument. If necessary, the working directory can be used to store data to not send out the same progress report twice. The progress report can be sent after each enclave invocation, if it lead to some progress.

Arguments:

- Import directory: where to look for the H data files and the metadata files
- Output directory: where to store finished reports
- What to do with older H files: delete or keep
- Another program to call to provide a progress report.
- Path to the HI client yaml conf file

3.1.3.3 Analytics Engine as a Finite State Machine

The executing code inside the analytics enclave is structured as a State Machine where a periodic external call is used to do processing incrementally, see Figure 3.

3.1.4 Two Different Calculation Scenarios

If the service is running, then input data becomes available to the Solution continuously and the aggregations over individual subscribers are performed accordingly. Final aggregation calculations are performed only infrequently at the end of each report period.

Alternatively, one might need to recalculate the stream of H for past data, possibly due to any problems in system operations. In such a case, the data warehouse in the MNO-VAD could act as a data repository and the stream could be replayed. Exported H files could also be used as a replay buffer to reduce DWH load. In that case, disk space must be allocated and run script modified accordingly to store relevant H files for recalculations.

The Solution supports both scenarios. The exact mechanism for feeding data into the Solution in the replay scenario is highly installation specific and is out of scope.

3.1.5 Mapping Requirements to Functionalities

The Solution is developed using the Sharemind HI platform. As such, some of its components are provided by the platform itself and described in the Solution analysis document. These components are the *Sharemind HI server* and *Sharemind HI client application*.

Figure 4 illustrates the mapping of functional requirements to the Solution components.

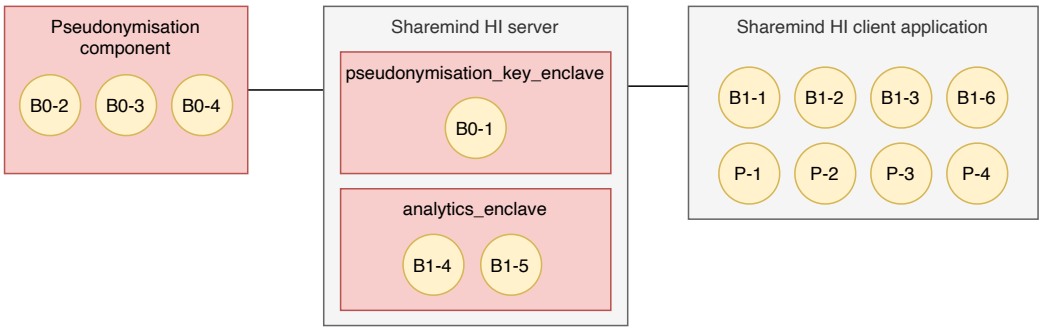

Figure 4. Mapping of functional requirements to the Solution components. Components shown in grey are provided by the Sharemind HI platform, components shown in red are to be designed and implemented as part of the Solution.

3.2 Sharemind HI Client Application

The requirements list a variety of interactions of authorised users with the Sharemind HI server such as:

- validating a *Dataflow Configuration*,
- starting tasks,
- providing inputs to topics,
- receiving and decrypting outputs of topics,
- auditing of computation results and logs.

Authorised users are specified in the *Dataflow Configuration* file and are identified by certificates that are signed by the Coordinator.

A component that needs to interact with the Sharemind HI server can use the Sharemind HI client library or the command-line Sharemind HI client application. The Solution uses the command-line client application. Components interacting with the Sharemind HI server, such as the pseudonymisation component, will execute the Sharemind HI client application with the appropriate parameters.

For example the following Sharemind HI client application command adds input datafile to the topic named “input”:

```
$ sharemind-hi-client -c config.yaml -a dataUpload -- --topic input
--datafile /path/to/datafile
```

The server configuration is specified in `config.yaml`. Sharemind HI client also performs remote attestation of the core and attestation enclaves before this action, in order to trust the Sharemind HI server. The server uses the client certificate to verify if the client is authorised to add data to the topic. The Sharemind HI client documentation can be read using `sharemind-hi-client --help`.

3.3 Task Enclaves

Intel SGX is a set of CPU instructions for creating and operating with enclaves. When an application creates an enclave, it provides a protected memory area with confidentiality and integrity

guarantees. These guarantees hold even if privileged malware is present in the system, meaning that the enclave is protected even from the operating system that is running the enclave. With enclaves, it is possible to significantly reduce the attack surface of an application.

When building an SGX application, the application should be separated into a trusted and untrusted part. The trusted part takes care of handling the private information and should be small. The untrusted part is responsible of orchestrating the rest of the application. It creates the enclaves, reads and writes files, communicates over the network and carries out processing on data that is not privacy sensitive.

Sharemind HI deployments use a core enclave, key enclave, attestation enclave and a number of task enclaves. The core, key and attestation enclaves are necessary for secure inner operation of Sharemind HI platform while task enclaves contain the components of the application that handle private information. In the Solution, functionality B0-1 is implemented as task enclave called *pseudonymisation_key_enclave* and B1-4 and B1-5 are implemented as *analytics_enclave*.

Sharemind HI tasks can read inputs and write outputs to data queues called topics. Inputs are added and outputs are downloaded using the Sharemind HI client application. Inputs and outputs are stored on disk and are encrypted with a key that is not known to the host system, so that the data can be decrypted only inside the enclave or by authorised clients. Sharemind HI also allows storing intermediate results used by the task as encrypted files that are not readable outside of the enclave.

3.3.1 Data Analysis Process Implemented by the Task *analytics_enclave*

This section gives an overview of the data analysis process that the task enclave implements. The analysis process is described in more detail in Annex 7.1.

The goal of the analysis is to calculate reports describing the geographical distribution of the population over some time period. The overall process is illustrated in Figure 5.

The national territory[4] has been divided into 1 km $\times$ 1 km grid units (hereinafter called “tiles”). The timeline of mobile events has been divided into 24-hour periods with optional configurable offset in this PoC implementation. The implementation is compatible with the following setups:

- MNO uses UTC time in positioning data.
- MNO uses local time in positioning data; the time conversion is done before pseudonymisation processing.

Subscriber data in the MNO-VAD system has been pseudonymised so that every subscriber has a different pseudonym for each period using the mechanism described in section 3.5.

Each pseudonymisation period is assigned a integer index. Index i_p is defined as count of days since Jan 1st, 1970 to beginning of period p:

$$i_p = start_p/(24 \cdot 3600)$$

where $start_p$ is Unix epoch (seconds) of period p start time.

[4]To take into account roaming subscribers, a separate tile is created for each foreign country (e.g. the whole Spain would be seen as a single tile by an operator in France).

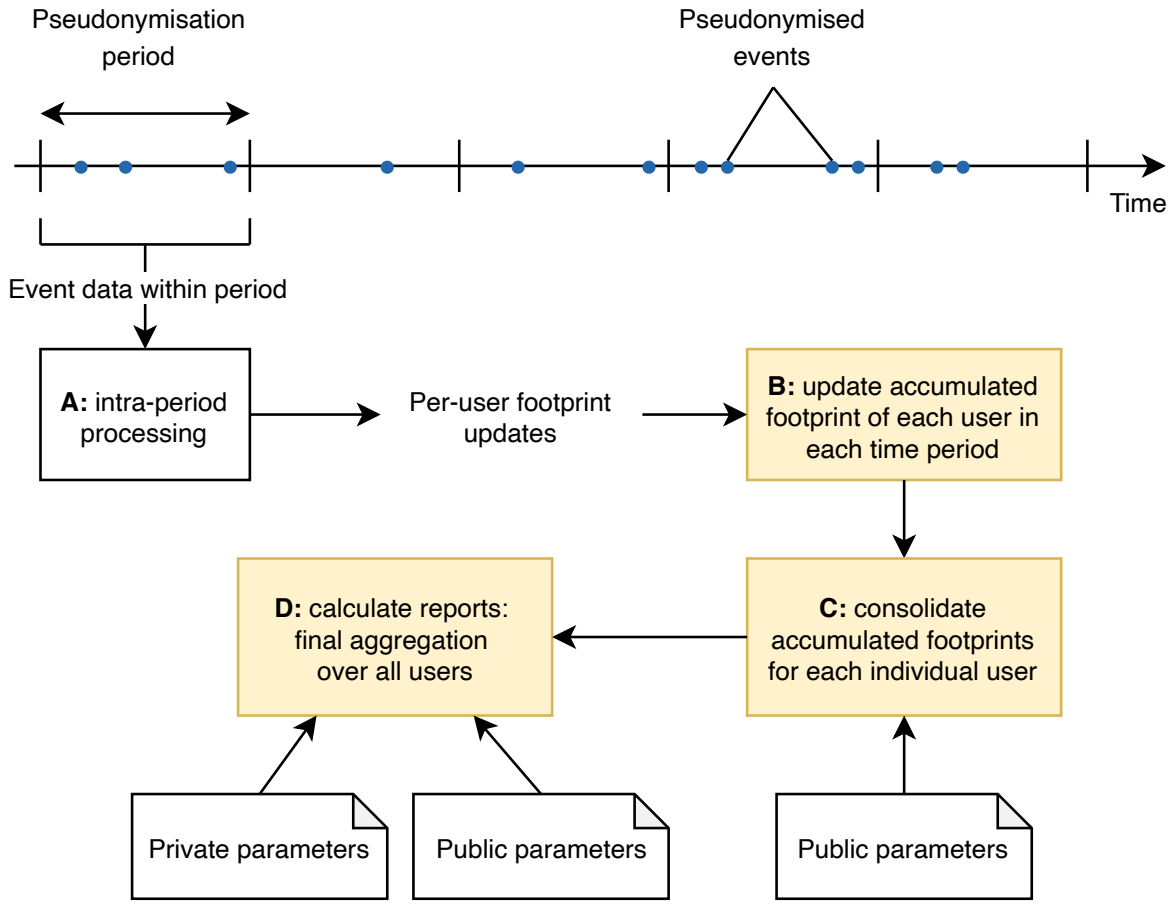

Figure 5. Data analysis process with modules. Yellow indicates components implemented in the Solution. The input data is pseudonymised before module A.

The MNO-VAD has implemented module A which pre-processes events in a period using conventional software. Module A calculates a matrix for each subscriber where an element of the matrix represents the frequency of observation in a tile during the day. We call these matrices subscriber footprint updates. It is assumed that the mobile positioning data timestamps are not changed from pseudonymisation to Module A processing.

The goal of module B is to calculate accumulated subscriber footprint matrices from the outputs of module A. The accumulated footprint elements represent the frequency of observation in a tile over a longer period. This module requires reversing the periodic pseudonyms in order to link subscriber data across periods.

Module C calculates a summary of the accumulated footprint for each subscriber. The summary is a binary matrix designating which tiles are in a subscriber's usual environment.

Module D aggregates the summaries of all subscribers into reports. Module D is run infrequently to calculate reports over a longer period of time such as six months. Statistical disclosure control is applied to both reports before making them available to either NSI and MNO-VAD. Note that the outputs of module D are aggregated and do not contain per-subscriber information.

Three reports are calculated by module D:

- The fingerprint report indicates for each tile how many subscribers have this tile in their usual environment.

- The top anchor distribution report indicates the spatial distribution of the subscribers' main place of living.
- The Functional Urban Fingerprint (FUF) report estimates the area of the commuting zones of cities using the calculated footprints and calibration data from NSI.

There are a few benefits to the proposed modularisation:

- Different reports can be added by changing module D without changing the other modules. Multiple analyses can take advantage of subscriber footprint accumulation.
- The system provides flexibility for deciding when to run different components. For example, subscriber footprint updates can be provided to module B every hour. This allows modules C and D to do less work during analysis by spreading out some of the computation overhead over a longer period. If necessary, the whole process can also be run as a single monolithic task by providing all of the footprint updates that have been generated during the analysis period to module B one-by-one when the analysis is started.

The *analytics_enclave* implements modules B, C and D. Module A is implemented as a separate deliverable.

3.3.2 Handling Large Amounts of Data

The geographical footprint data processed by the Solution exceeds the limited amount of memory available to an Intel SGX enclave. Current SGX processors support up to 128 MB of protected memory (*Enclave Page Cache*) but some of this is reserved for the SGX platform itself. Larger amounts of memory can be used by paging to disk. While paging is done automatically it can have heavy performance cost depending on data access patterns. Algorithms that are heavy in random accesses can operate an order of magnitude slower than unprotected code.

Even if page faults were minimised, the paging functionality still limits the amount of data that can be processed to the amount of memory available. Another option is to use storage as extended memory. By encrypting stored data using a key that is only available in the enclave it is possible to handle larger-than-memory datasets.

For efficient sequential processing of large data, Sharemind HI has an API for stream programming. A stream is a sequence of values. There are three types of streams: sources that produce values, pipes that transform values and sinks that consume values. Sources, pipes and sinks can be composed. For example, to count the number of elements in a large file using a bounded amount of memory one can combine a source that produces a stream of elements from a file and a count sink. Pipes may store intermediate values in encrypted temporary files. For example, this is necessary for sorting data. The streaming API can be used to implement efficient programs on large datasets.

The *analytics_enclave* design uses both the paging support of SGX and the streaming API of Sharemind HI. Streams allow *analytics_enclave* to process data that does not fit into memory. Paging allows the enclave to use more than 128 MB of memory while processing a chunk of a stream. The algorithm implementation will process stream chunks sequentially if possible to minimise paging.

3.3.3 Access Control

The Solution requires access control functionality to ensure that only authorised parties can provide inputs, run tasks and retrieve outputs. The Solution uses a modular design where data and services are grouped into two separate enclaves. The access to each task enclave is controlled in Sharemind HI through topics and roles – runner, input producer for topic, output consumer for topic. The roles are defined by the Dataflow Configuration file which is part of the Sharemind HI solution.

3.3.3.1 Example of Sharemind HI Dataflow Configuration

In system the Dataflow Configuration is provided as text file. On Figure 6 same dataflow is depicted graphically.

```
Stakeholders:
    - Name: "ND"
      CertificateFile: "..."
    - Name: "VAD"
      CertificateFile: "..."
    - Name: "NSI"
      CertificateFile: "..."
Auditors:
    - "(stakeholder name)"
Enforcers:
    - "ND"
    - "VAD"
    - "NSI"
Tasks:
    - Name: "pseudonymisation_key_enclave"
      EnclaveFingerprint: "..."
      SignerFingerprint: "..."
      Runners:
        - "ND"
    - Name: "analytics_enclave"
      EnclaveFingerprint: "..."
      SignerFingerprint: "..."
      Runners:
        - "VAD"
Topics:
    - Name: "nsi_input"
      Producers:
        - "NSI"
      Consumers:
        - "analytics_enclave"
    - Name: "periodic_pseudonymisation_key"
      Producers:
        - "pseudonymisation_key_enclave"
      Consumers:
        - "pseudonymisation_key_enclave"
```

```
    - "analytics_enclave"
    - "ND"
- Name: "fingerprint_report"
  Producers:
    - "analytics_enclave"
  Consumers:
    - "VAD"
    - "NSI"
- Name: "top_anchor_distribution_report"
  Producers:
    - "analytics_enclave"
  Consumers:
    - "VAD"
    - "NSI"
- Name: "functional_urban_fingerprint_report"
  Producers:
    - "analytics_enclave"
  Consumers:
    - "VAD"
    - "NSI"
- Name: "statistics"
  Producers:
    - "analytics_enclave"
  Consumers:
    - "VAD"
    - "NSI"
- Name: "application_log"
  Producers:
    - "analytics_enclave"
  Consumers:
    - "NSI"
    - "VAD"
```

The runners of each enclave are configured explicitly. Consumers and producers are defined via topics.

The Dataflow Configuration file describes data flow in the Solution as a set of “topics”, where each topic carries certain kind of information identified by the topic name. Each topic has one or more producers and one or more consumers. Producers and/or consumers can be external stakeholders or the task enclaves in the Sharemind HI solution instance.

3.3.4 Task Enclave Input and Output Parameters

This section describes the format of the inputs and outputs of the task enclaves.

There are two methods for providing inputs to task enclaves in Sharemind HI:

- Uploading a task input to a topic before running the task. Inputs are stored as encrypted files on disk. The enclave code can access the decrypted inputs.

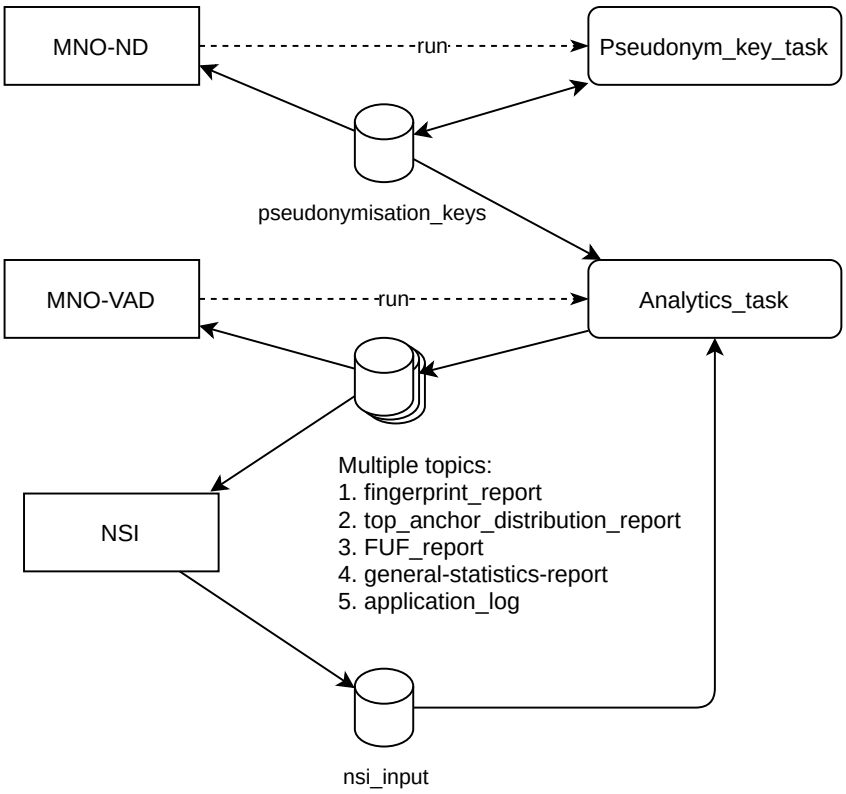

Figure 6. Dataflow configuration graph

- Passing a parameter to the task when running it. Parameters are represented as strings. Parameters are visible to all authorised users of Sharemind HI server.

Topics are designed for input data, while parameters are more suitable for public scalar values.

The *pseudonymisation_key_enclave* handles one process: generating periodic pseudonymisation keys, whereas *analytics_enclave* handles two processes: updating accumulated footprints and calculating reports.

The following subsections will describe inputs and outputs of the task enclaves.

3.3.4.1 *pseudonymisation_key_enclave*

The *pseudonymisation_key_enclave* generates a periodic pseudonymisation key. This task enclave implements business case requirement B0-1.

Inputs:

- `period` - integer index of the period for which the periodic pseudonymisation key is requested. Provided as a string literal parameter.

Outputs:

- `periodic_pseudonymisation_key` - periodic pseudonymisation key stored in the topic with the same name.

3.3.4.2 *analytics_enclave*

The *analytics_enclave* calculates accumulated footprints and reports. It implements business case requirements B1-4, B1-5 and modules B, C and D in the data analysis process. Each time that MNO-VAD runs *analytics_enclave* it should provide a footprints update file as a parameter. *analytics_enclave* then uses the algorithm in section 3.1.3.2, the input file and the NSI report request to decide whether to ingest the updates into accumulated footprints and whether to calculate reports after updating accumulated footprints.

Inputs:

- `nsi_input` - report request provided by the NSI in the topic with the same name. Its structure is described in section 4.2.7. It contains:
  - The report request containing the report period. The structure is described in section 4.2.4.
  - Reference areas used for functional urban fingerprint calculation. The structure is described in section 4.2.5.
  - Use case flag, for whether to use the following census data for calibration or not.
  - (optional) Census data for the number of residents in a tile. The structure is described in section 4.2.6.
- `file` - name of the footprint updates (H) file to be processed. Provided as a public string parameter by the MNO-VAD when running the enclave. The structure of the footprint updates file is described in section 4.2.3.
- `period` - period of the footprint updates (H) file to be processed. Provided as a public string parameter by the MNO-VAD when running the enclave.
- `finish-report` - provided as a public flag parameter by the MNO-VAD when running the enclave (value irrelevant). When provided as the only parameter, the report will be generated and thus the current report request fulfilled.
- `cancel` - provided as a public flag parameter by the MNO-VAD when running the enclave (value irrelevant). When provided as the only parameter, the current report request will be canceled.

Outputs:

- `fingerprint_report` - fingerprint report. Stored in the topic with the same name. The structure is described in section 4.3.1.
- `top_anchor_distribution_report` - report of the spatial distribution of the subscribers' "main place of living". Stored in the topic with the same name. The structure is described in section 4.3.3.
- `functional_urban_fingerprint_report` - functional urban fingerprint report. Stored in the topic with the same name. The structure is described in section 4.3.2.
- `statistics` - additional statistical values. Stored in the topic with the same name. The structure is described in section 4.3.4.
- `application_log` - human readable information about the state of the enclave. It also contains the indicators, which are described in section 6.2. More information is available in section 3.3.5.

This process also reads and writes the accumulated footprints table which is described in section 4.4.1.

3.3.5 Application Log for the Activities in the Task Enclaves

Sharemind HI task enclaves, carrying out statistical calculations, produce a human-readable application-specific log. Application log records all API calls to the server and also start and end of all file activities. In addition, it contains sanity checks and non-sensitive metadata about the data processed in the enclave (e.g., the number of distinct pseudonyms given as input for each period and for each mobile network/country code value). This information can be later used for debugging purposes and, if necessary, to add new checks into next iterations of the analysis code. It is possible to increase or decrease the logging verbosity in future versions of the Task enclaves.

Moreover, the Sharemind HI server itself will keep an encrypted and hash-chained audit log. Auditors can decrypt parts of these logs to carry out auditing and ensure that the system has not been compromised in any way. The audit log does not record any information about the processed data or results, beyond relative sequence order of performed Sharemind HI level API calls that include also dataflow topics read/write operations.

3.3.6 Persistent Data

The *analytics_enclave* must store values between invocations of tasks:

- the accumulated geographic subscriber footprints,
- the top anchor distribution.

Sharemind HI provides an API for securely storing data outside of the enclave. The data is encrypted using a key that is only available to the enclave. This means that the untrusted host machine can not read or modify these values. Therefore, for example, none of the involved organisations (MNO, NSI, external auditors etc.) can access the accumulated subscriber footprints. The data format of accumulated subscriber footprints is provided in section 4.4.1 and the format of the top anchor distribution is given in section 4.4.2.

3.4 Pseudonymisation Component

The pseudonymisation component is part of the MNO-ND information system and handles periodic pseudonymisation of MNO-ND data. The pseudonymised data is then transmitted to the MNO-VAD for other kinds of analysis, e.g. commercial analytics and other kinds of non-official statistics.

In a production system it would be reasonable to implement the pseudonymisation component as a library. Since different MNOs may use different software architectures the pseudonymisation component is implemented as a REST service in the PoC which allows easy integration. It uses the Sharemind HI client application to communicate with the Sharemind HI server. The integration of the pseudonymisation component with the MNO-ND information system is illustrated on Figure 7.

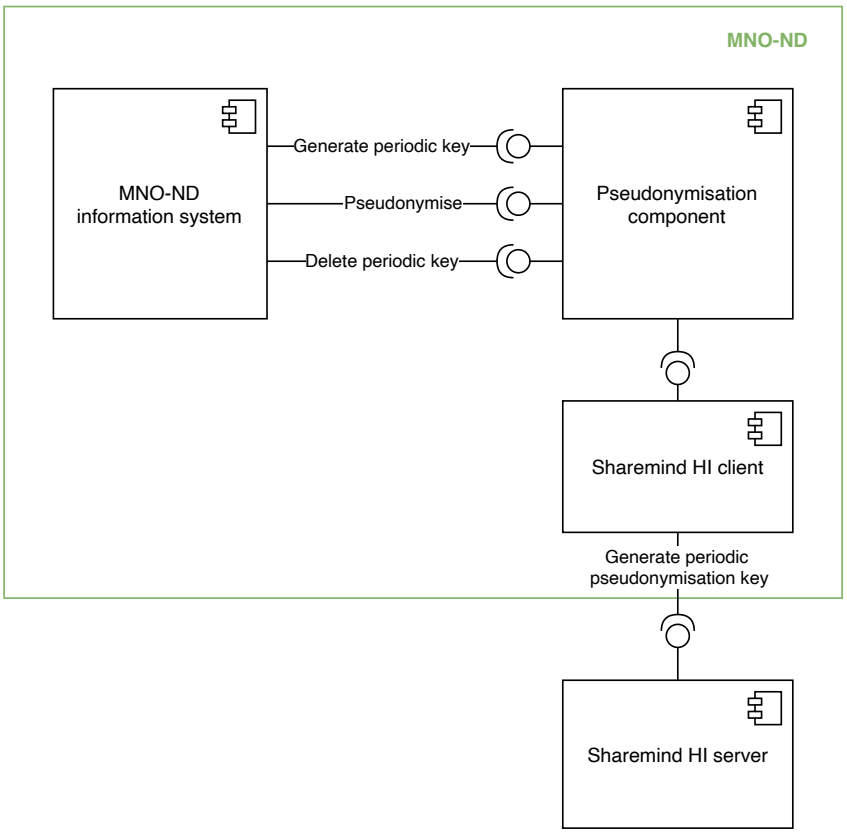

Figure 7. Pseudonymisation component's integration with the MNO-ND information system.

To generate a periodic pseudonymisation key a `POST` request is sent to the `/v1/key/{period}` endpoint where `period` is a date in YYYY-MM-DD format. The pseudonymisation component retrieves the periodic key from the *pseudonymisation_key_enclave* and keeps this key in memory. After pseudonymisation of IMSIs in a period, the key is deleted using a `DELETE` request to the `/v1/key/{period}` endpoint.

To pseudonymise data the IMSIs should be sent as a JSON payload to the `POST` endpoint `/v1/pseudonymise` with the following structure:

Field	Type	Description
period	string	Pseudonymisation period date in YYYY-MM-DD format
identifiers	array[string]	Array of IMSIs to pseudonymise

The pseudonymisation component responds with either of the following:

a) status code 400 and the following JSON payload, in case of errors:

Field	Type	Description
message	string	Error message

b) or with status code 200 and the following JSON payload, if pseudonymisation succeeded:

Field	Type	Description
pseudonyms	array[string]	Array of base64 encoded pseudonyms

The pseudonymisation component uses the Sharemind HI client application to communicate with the Sharemind HI server to generate the periodic pseudonymisation key. The enclave parameters of the periodic key generation process are described in section 3.3.4.

IMSIs can be pseudonymised in batches or one by one as is suitable for the MNO-ND information system. The process for pseudonymising IMSIs for a period is thus:

1. Retrieve periodic pseudonymisation key using a `POST` request to `/v1/api/key`.
2. Pseudonymise IMSIs using `POST` requests to `/v1/pseudonymise`.
3. Delete periodic pseudonymisation key using a `DELETE` request to `/v1/api/key`.

Note that while the pseudonymisation component is implemented as a service it should be considered as a module of the MNO-ND system. To mitigate possible network-level risks (e.g. man-in-the-middle), both the server and client side of the pseudonymisation component should be deployed inside the MNO-ND, preferably in a single machine (the service should be configured to only listen for incoming connections from local machine and not over an actual network). HTTP is only used as an interface because of its good support in different programming languages.

3.5 Pseudonymisation Scheme

In this section we will describe the abstract pseudonymisation scheme. The proposed scheme using Sharemind HI allows the MNO-ND to pseudonymise data using periodic keys such that the long-term pseudonyms can be reconstructed in the task enclave for longitudinal analysis.

The *pseudonymisation_key_enclave* generates periodic pseudonymisation keys which are 128-bit random sequences. Let K_i designate the periodic pseudonymisation key of period i.

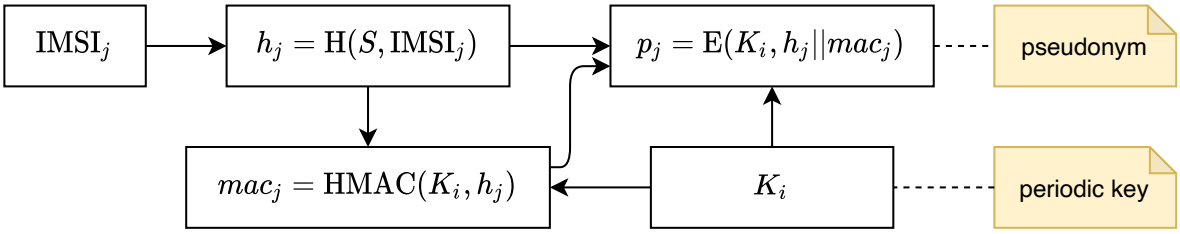

Figure 8. Simplified diagram of periodic pseudonymisation.

The pseudonymisation process implemented by the pseudonymisation component is as follows (illustrated on Figure 8):

1. The input event records are in the form $(\mathrm{IMSI}_j, t_j, f_{j,1}, \ldots, f_{j,n})$ where IMSI_j is the identifier, t_j the event timestamp and $f_{j,1}, \ldots, f_{j,n}$ are fields describing the event.
2. Let i denote the period that timestamp t_j belongs to. Retrieve periodic pseudonymisation key K_i for period i from the *pseudonymisation_key_enclave*. The periodic pseudonymisation key is transmitted to the pseudonymisation component over a secure channel.

3. Let $\mathrm{E}(K, x)$ designate the AES-128 encryption function with key K and input x. Let $\mathrm{H}(S, x)$ designate the SHA-256 hash function with input x and salt S that can be chosen by MNO-ND. Let $\mathrm{HMAC}(K, x)$ designate HMAC-SHA256 with key K and input x. Let m, n be integers such that $m + n = 128$. Let $x_{0..n-1}$ designate the first n bits of x. Let $a \| b$ designate the concatenation of bytestrings a and b.
   a) Hash the IMSI using $h_j = \mathrm{H}(S, \mathrm{IMSI}_j)_{0..m-1}$. The identifier is one-way hashed to supply an additional layer of protection. This is not strictly necessary since only the enclave can see IMSIs during the analysis.
   b) Calculate MAC from the IMSI using $mac_j = \mathrm{HMAC}(K_i, h_j)_{0..n-1}$. The MAC is used to check whether the pseudonyms were constructed correctly when the pseudonyms are decrypted before analysis.
   c) Calculate the pseudonym p_j by encrypting the hash and MAC using $p_j = \mathrm{E}(K_i, h_j \| mac_j)$.
4. Let mcc_j and mnc_j designate the mobile country code and mobile network code of the IMSI of record j. The pseudonymised event record $(p_j, t_j, mcc_j, mnc_j, f_{j,1}, \ldots, f_{j,n})$ is output in a way that is suitable for the given MNO value-added services department (stored in a database, pushed to an event queue, etc). MNO-VAD department is free to perform any analytics allowed by contracts and regulations.

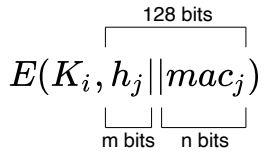

Figure 9. Truncation of encryption inputs.

The block width of AES is 128 bits. To fit both the hashed IMSI and the HMAC tag in the periodic pseudonym some bits are truncated from both. To choose appropriate values of m and n we should analyse the possibility of collisions. When picking n values from a uniform distribution with N values ($n \leq N$) the probability of a collision is:

$$P(N, n) = 1 - \frac{N!}{N^n (N-n)!} \approx 1 - \exp(\log \Gamma(N+1) - \log \Gamma(N-n+1) - n \log N)$$

If for example there are 100 million IMSIs and $m = 96$ this gives a collision probability of $P(2^{96}, 10^8) \approx 6.3 \cdot 10^{-14}$. Note that this is the probability of collision in at least one pair among 100 million generated pseudonyms, not the probability of collision when generating two pseudonyms. The following table provides collision probabilities for some values of m and n.

m	n	$P(2^m, 10^8)$
96	32	$\approx 6.3 \cdot 10^{-14}$
104	24	$\approx 2.5 \cdot 10^{-16}$
112	16	$\approx 9.6 \cdot 10^{-19}$
120	8	$\approx 3.8 \cdot 10^{-21}$

The HMAC is used for an integrity check when the pseudonyms are reversed in the enclave. For example, the check will fail if the long-term pseudonyms were encrypted with the key of the wrong period. It is only designed to provide protection against mistakes so it is not critical that a significant amount of space is reserved for the HMAC. The m parameter will be a configuration option in the Solution. The authors recommend to pick one of $m = 96, 104, 112$ bits. $m = 96$ will be used in this document.

Note that in this case SHA-256 was picked as the hash function. This choice is not fixed. MNO-ND can use any other scheme for long-term pseudonymisation as long as the pseudonym fits in a 128-bit AES block. It is advised to leave some bits for the HMAC integrity check as well.

This design allows the enclave to reverse the encryption and retrieve the hashed IMSIs for analysis since the enclave knows the periodic pseudonymisation keys. The decrypted, but hashed, IMSIs are consistent across different periods and can then be used by the enclave for analyses spanning longer periods than the pseudonymisation period.

4 Data Description

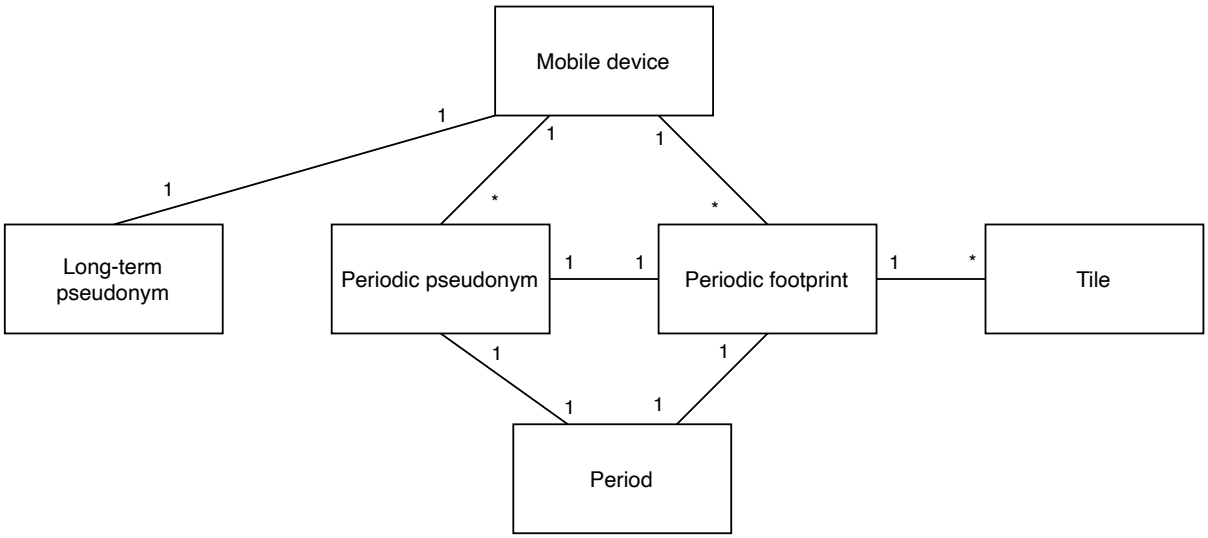

Figure 10. Conceptual data model.

This section describes the data elements used in the Solution as inputs, outputs or persistent internal data. Structured data is represented in a tabular format. H input files are read in as tabular binary files with extension .hdata. Other input and output data are stored as CSV. All persistent data in the task enclave are serialised as encrypted tabular binary files.

The tabular binary format can be read and streamed efficiently in Sharemind HI task enclaves. A tool is used to convert between CSV files and the binary format. This means that input data is stored as CSV files which are converted to the binary format before adding the inputs in Sharemind HI topics. CSV is a useful input format because it supported by different database systems and tools. The volume of H files is so large that direct creation in .hdata format by Module A was chosen in the Solution.

While hdata and CSV files will be used in the PoC, future versions of Sharemind HI will support Apache Parquet [5] files. Parquet is a columnar storage format that was designed for the Hadoop big data ecosystem and is more space-efficient than the current binary format. Parquet has also become popular in the Python data analysis ecosystem and there are tools (such as Apache Sqoop [6]) for exporting Parquet files from relational databases.

Basic metadata will be stored in the names of the files. In the following sections we will use `{x}` to represent variable x in filename descriptions. That is, "`filename_{date}.hdata`" denotes

[5] Apache Parquet: `https://parquet.apache.org/`
[6] Apache Sqoop: `https://sqoop.apache.org/`

filenames such as "`filename_2020-09-01.hdata`". Dates are represented in YYYY-MM-DD format.

All of the inputs are added to Sharemind HI topics, except for the subscriber footprint updates. Topic elements are stored on disk before the task is run and are encrypted such that the data is accessible in Sharemind HI enclaves, but not from the host system. Since the pseudonymised subscriber footprints data is controlled by MNO-VAD and the Sharemind HI server is also hosted by MNO-VAD encrypting the subscriber footprints is not necessary. Therefore, footprint updates are stored in an unencrypted tabular binary format and *analytics_enclave* reads them from disk.

4.1 Overview of Data in the Solution and Its Life Cycle

Data	What creates	What initiates deletion	Who can view	Who can delete
H	cron job	processed by analytics enclave	MNO-VAD	MNO-VAD sysadmin
report request	NSI analyst upload	not deleted	NSI, MNO-VAD	NSI analyst
RA	NSI analyst upload	not deleted	NSI, MNO-VAD	NSI analyst
L	NSI analyst upload	not deleted	NSI	NSI analyst
D', P', C	report calculation is ready	not deleted	NSI, MNO-VAD	NSI analyst
S	cron job/report calculation	report calculation is ready, enclave update	nobody	report calculation app
P and other temp data	report calculation app	at end of calculation	nobody	not applicable

4.2 Input Data

Input data files are represented in a tabular format as CSV files. A tool is used to convert CSV files into a binary tabular format before the inputs are added to Sharemind HI topics.

This section describes the structure and filenames of input data objects. The symbolic name for each data entity is provided in the section title in parentheses.

4.2.1 Geographical Coordinates of the Tiles

The geographical coordinates of the tile are given as easting and northing. For file size optimisation the coordinate unit is 1km (size of tile). Standard projected coordinates with 1m unit can be obtained by multiplying the value by factor 1000 m/km.

4.2.2 Periodic Updates of Subscriber Footprints from MNO-VAD (H) in CSV Format

This data format could be used by MNO-VAD integration in combination with CSV-to-binary conversion utility. In PoC this format is not used on MNO-VAD site. Therefore the names of such files are not specified.

Column name	Type	Description
id	16 bytes in base64	Subscriber identifier
tile_e	unsigned integer	Geographic grid coordinates easting
tile_n	unsigned integer	Geographic grid coordinates northing
value_0	floating point number	Value of the score in this tile in sub-period 0
value_1	floating point number	Value of the score in this tile in this sub-period 1
value_2	floating point number	Value of the score in this tile in this sub-period 2
value_3	floating point number	Value of the score in this tile in this sub-period 3

4.2.3 Periodic Updates of Subscriber Footprints in Sharemind HI Binary Format (H)

This data is created daily by MNO-VAD export.

Column name	Type	Description
id	`uint8[16]`	Subscriber identifier
tile_e	`uint16`	Geographic grid coordinates easting
tile_n	`uint16`	Geographic grid coordinates northing
value_0	`float32`	Value of the score in this tile in sub-period 0
value_1	`float32`	Value of the score in this tile in sub-period 1
value_2	`float32`	Value of the score in this tile in sub-period 2
value_3	`float32`	Value of the score in this tile in sub-period 3

Footprint update files are named `day-{DATE}-update.hdata` where {DATE} is formatted as YYYY-MM-DD. DATE is start date of pseudonymisation period, formatted as UTC timezone, i.e. without timezone offset.

Each data file shall be accompanied with metadata file with same name but with extension .hdata.meta. Metadata file contains human-readable info about file processing, to be included into application-data logging.

4.2.3.1 Handling of Invalid Records in Periodic Updates of Subscriber Footprints (H)

The value fields of each H record must be non-negative and at least one sub-period must have a positive value. If a record has a negative value or the maximum of the values is 0 the record is skipped.

By spec each record in H file shall have unique <id, tile_e, tile_n> combination. If n records have the same <id, tile_e, tile_n> combination in same H file then $n - 1$ records are considered

duplicates. If duplicate records are present then the data is still processed but the results are less reliable. Records with the same key are combined into one by setting the value of sub-period i to the maximum of sub-period i values of the duplicates.

4.2.4 Report Request

Report requests are not tabular data so they are not represented as such. A report request consists of the start and end date of the report period. These dates are represented as Julian day numbers using 32-bit unsigned integers. The two integers are serialised as a packed sequence of bytes in this order: start date, end date.

4.2.5 Reference Areas (RA)

Column name	Type	Description
id	unsigned integer	Reference area index
tile_e	unsigned integer	Easting coordinate of geographic tile that is part of the reference area
tile_n	unsigned integer	Northing coordinate of geographic tile that is part of the reference area

The reference areas file is named "`reference_areas_{date}.csv`".

4.2.6 Census Data for Absolute Number of Residents in a Tile (I)

Column name	Type	Description
tile_e	unsigned integer	Geographic grid coordinates easting
tile_n	unsigned integer	Geographic grid coordinates northing
value	floating point number	(Estimated) number of residents in the tile

The census resident data file is named "`census_residents_{date}.csv`".

4.2.7 Binary NSI Input

The NSI input combines the report request, the reference areas, use case and census data.

Column name	Type	Description
first_period	`uint32`	Part of the report request
last_period	`uint32`	Part of the report request
with_calibration	`uint64`	Boolean describing the use case
num reference area rows	`uint64`	The number of valid reference area rows
reference area rows	`RA[1000000]`	Space for reference area rows
num census resident rows	`uint64`	The number of valid census resident rows
census resident rows	`l[1000000]`	Space for census resident rows

The binary NSI input has a fixed size and can easily be represented as a C `struct`.

4.3 Output Data

4.3.1 Fingerprint Report (D')

Column name	Type	Description
tile_e	unsigned integer	Geographic grid coordinates easting
tile_n	unsigned integer	Geographic grid coordinates northing
value_0	floating point number	Estimated number of subscribers with this tile in their usual environment in sub-period 0
value_1	floating point number	Estimated number of subscribers with this tile in their usual environment in sub-period 1
value_2	floating point number	Estimated number of subscribers with this tile in their usual environment in sub-period 2
value_3	floating point number	Estimated number of subscribers with this tile in their usual environment in sub-period 3

4.3.2 Functional Urban Fingerprint Report (FUF aka C)

Column name	Type	Description
reference_area	unsigned integer	Reference area index
tile_e	unsigned integer	Geographic grid coordinates easting
tile_n	unsigned integer	Geographic grid coordinate northing
strength	floating point number	Connection strength of the tile to this reference area

4.3.3 Top Anchor Distribution Report (P')

Column name	Type	Description
tile_e	unsigned integer	Geographic grid coordinates easting
tile_n	unsigned integer	Geographic grid coordinates northing
count	unsigned integer	Number of users for whom this tile was the top anchor tile

4.3.4 Statistics

Column name	Type	Description
highly_nomadic_users	unsigned integer	Number of subscribers who did not have tiles after footprint quantisation
observed_total_users	unsigned integer	Total number of observed subscribers
adjusted_total_users	floating point	Total number of observed subscribers after

Column name	Type	Description
	number	adjustment

4.4 Internal Persistent Data

This section describes the format and filenames of internal persistent data. Accumulated subscriber footprints and the fingerprint report are stored in a tabular format.

4.4.1 Accumulated Subscriber Footprints (S)

Column name	Type	Description
id	`uint8[12]` [7]	Subscriber identifier
tile_e	`uint16`	Geographic grid coordinates easting
tile_n	`uint16`	Geographic grid coordinates northing
value_0	`float32`	Value of the accumulated score in this tile in this sub-period 0
value_1	`float32`	Value of the accumulated score in this tile in this sub-period 1
value_2	`float32`	Value of the accumulated score in this tile in this sub-period 2
value_3	`float32`	Value of the accumulated score in this tile in this sub-period 3

The accumulated subscriber footprints file is named "`accumulated_footprints_{date}.bin`".

4.4.2 Top Anchor Distribution (P)

Column name	Type	Description
tile_e	`uint16`	Geographic grid coordinates easting
tile_n	`uint16`	Geographic grid coordinates northing
count	`uint32`	Number of subscribers for whom this tile was the top anchor tile.

The top anchor distribution file is named "`top_anchor_distribution_{date}.bin`".

[7] The length of the long-term pseudonym depends on the parameter m which is described in section 3.5. In this document $m = 96$ bits which is 12 bytes.

5 Hardware

5.1 Hardware Considerations

This PoC will use a single Sharemind HI computing node. The hardware specification will be based on assumptions given in section 2.2.3.1, above.

5.1.1 CPU

The Solution must be deployed on a CPU with Intel SGX support and correct configuration according to the accompanying manuals. Enclave calculations are currently performed in a single thread as multithreading in enclave increases the attack surface for possible vulnerabilities.

Due to several published security vulnerabilities (e.g. *Meltdown*, *Spectre*, *Load Value Insertion*), older generations of SGX-enabled CPUs require some CPU microcode patches that also have a negative effect on performance. For newer generations, these vulnerabilities have been mitigated by hardware design, which should boost both their performance and SGX platform security at the same time.

5.1.2 RAM

There is a limited amount of SGX-protected memory (*Enclave Page Cache*, typically 128 MB) available in each CPU and this amount is shared by all enclaves. The designed Solution uses more memory and this requirement is satisfied by using SGX paging cache (EPC). EPC provides the same security guarantees as SGX but at reduced performance. Performance loss depends on memory access patterns of the enclave software. In the current Solution the EPC performance is satisfactory.

5.1.3 Disk

The Solution uses large volumes of sequential disk IO. Therefore, hardware must provide good sequential throughput. Striped RAID SSD with throughput of 800MB/s is assumed in scalability calculations.

The following persistent data is stored on disk:

- H data for the whole report period – ca 12TB
- input files uploaded by NSI – less than 0.1TB
- ready downloadable reports – less than 0.1TB
- optional: incrementally calculated S data – ca 1.4TB

Also temporary disk space is needed during the calculations, ca 4TB.

5.2 Hardware Resource Requirements

The Solution requires

- 16TB of disk space. Local SSD drives in RAID 10 configuration are recommended for optimal performance.
- 16GB RAM.
- Intel CPU with SGX capabilities, with relatively high clock, at least 4 cores.

Note: the required disk space depends on the volume of involved MNO subscribers and on the methodology of footprint calculation.

6 Plan for Testing and Quality Assurance

6.1 Overview of Tests

Following test activities are performed

- validate correctness and unambiguity of statistical algorithms description
- validate the functionality of Sharemind HI solution code
  - validate implementation of statistical algorithm in Sharemind HI solution code
  - business-cycle tests covering main usecases
- load and scalability tests

Tests are described and testing results are discussed in deliverable “Testing Report”.

6.2 Indicators for Data Quality Validation

During processing several indicator values are calculated to characterise the data and validate the data quality. The goal of data validation is to detect gross errors or intentional data manipulation in earlier stages of data collection and preprocessing. The indicators’ values per se do not guarantee data quality. The changes in indicator values might reveal disturbancies in data processing. Indicator value dynamics must be evaluated by human expert to estimate data corruption risk.

It is assumed that detailed analysis of data applicability for the purpose of statistical reports is already performed while producing input footprint data; and the indicators described here do not cover that analysis.

The result of the validation process is a report per each processed H file. The report is provided to the NSI and MNO-VAD as a download by means of Sharemind HI client application. The report contains the calculated descriptive statistics – the indicator values. The report shall implement SDC. The report could be used as an input for raising alarms in the Operations and Maintenance (O&M) system.

The data quality indicators are described below. The indicators mostly focus on H files. “H file” is used as a synonym for footprint data of all mobile subscribers for one pseudonymisation period, see chapter 4 “Data Description”.

6.2.1 Technical Metadata Helping to Trace Back During Troubleshooting

This information helps to understand where the data came from and helps to relate it to logs produced by previous steps. The following indicators are reported:

- H file metadata (full paths of binary and csv files as visible to process, sizes and filesystem dates, the user who started the processing, start and end of binary-to-csv conversion and any metadata provided by Module A alongside the H file),
- count of missing H files before this H (should be 0),
- count of H invocations ignored due to strict period monotonicity requirement(count should be 0).
- processing start and end time of H in HI server.

Examples: if H files are invoked in period order 1,2,2,3 then non-monotonic ignore count is 1 (second 2 is ignored). if period order is 1,9,2,3 then ignore count is 2 (2,3 are ignored)

6.2.2 Number of Duplicate Records in the H File

By specification, each record in the H file shall have a unique ‹id, tile_e, tile_n› combination. If n records have the same ‹id, tile_e, tile_n› combination then $n - 1$ records are considered duplicates. If duplicate records are present then the data is still processed but the results are less reliable.

One integer value is produced: count of duplicate records in file.

6.2.3 Unique Subscriber Counts and Record Counts in H and S Files

This information helps to detect if the data is partial, or large number of new subscribers are introduced (e.g. due to some problem with pseudonym continuity).

Indicators are calculated for each of following files:

- H
- S before processing this H
- S after procession this H

Indicators are:

- count of unique subscribers in file,
- count of records in file,
- histogram of count of records per subscriber in file (histogram bin boundaries are power of 2 1,2,4,8 . . . with a total of 10 bins).

In total 3 files * 3 indicators = 9 indicator values are calculated while processing a single H file. Of these, 6 indicator values are integer counts, 3 values are histograms.

6.2.4 Copresence of Subperiods in H Records

The following indicator helps to detect if some subperiod's data are missing or spatially not correlated with other subperiods or suspiciously highly correlated.

- Histogram: count of H records with given superiod pattern (subperiod order in pattern 0, 1, 2, 3). 0 in pattern position i means given subperiod i had weight 0 in given record, 1 means weight ›0.

A single histogram is calculated over all records in H.

Example of coding: the data present in subperiods 0, 1 and 3 would produce pattern “1101”.

Examples of results’ interpretation (x represents arbitrary value in given position in pattern):

- If codes 0xxx present then something suspicious (if subscriber present in given tile during subperiod x then should be also present in subperiod 0 that covers whole day)
- If codes x00x and x11x are strongly present and x10x and x01x are missing then correlation between subperiods 1,2 is suspiciously strong, i.e subperiods 1 and 2 are clones of each other and dont have any individual info which suggest a distortion in data.
- If codes x0xx are strongly present and codes x1xx are missing then people seem to be absent from subperiod 1. If codes x0xx are missing and codes x1xx are strongly present then people seem to be present in subperiod 1 in each subscriber each tile in H. Both cases are highly unlikely to happen with correct data.

6.2.5 Spatial Distribution of Data

Many data issues could be observed as an abrupt change of data volume and distribution. For example, apparent sudden increase in mobility of subscribers could be caused by problems with the continuity of pseudonyms over time or by the mobile network’s internal problems like MNO-ND switching to an invalid cellplan. A cellplan is the data describing the mobile network and specifying for each mobile antenna the location of working area a.k.a. cell.

The coordinates of a tile are <tile_e, tile_n>. It is assumed that these are projected centroid coordinates which are approximately Euclidean coordinates with unit 1 meter.

The indicators below are calculated separately for each subperiods [0..3]:

- in H, the histogram of count of unique tiles per subscriber where presence is not 0 (histogram bin boundaries are power of 2: 1,2,4,8 .. 512 with a total of 10 bins),
- in H, the histogram of weight values in given subperiod (histogram bin boundaries are power of 2 starting from 1/256,1/128 .. 256 with a total of 17 bins),
- in S, before processing this H the histogram of weight values in given subperiod (histogram bin boundaries are power of 2: 1/256,1/128 .. 256 with a total of 17 bins),
- the histogram of distance between subscriber average position in previous S and this H. Average position is calculated as the weighted average of tile coordinates in S records for the current subscriber and current day sub-period. (histogram bin boundaries are power of 2: 256,512 .. 131072 with a total of 10 bins). Abrupt day-to-day change in this indicator would indicate that something changed in data collection or in real movement (e.g start of vacation season). *Technical note: Comparing H with previous period H would be more sensitive and consistent indicator but that would involve more overhead.*
- histogram of subscriber tiles “bounding box” diagonal length in current H, using only the tiles where presence is not 0 in a given subperiod (histogram bin boundaries are power of 2: 1024,2048 .. 131072 with a total of 8 bins),
- histogram of subscriber tiles “bounding box” diagonal length in S before adding current H, using only the tiles where presence is not 0 in a given subperiod (histogram bin boundaries are power of 2: 1024,2048 .. 131072 with a total of 8 bins),
- histogram of subscriber tiles “bounding box” diagonal length difference between S before and S after processing, using only the tiles where presence is not 0 in a given subperiod (histogram bin boundaries are power of 2: 1024,2048 … 131072 with a total of 8 bins).

In this section, a total of $7 \cdot 4 = 28$ histograms are calculated while processing one H file.

Bounding box diagonal D for set of points $\langle x_i, y_i \rangle$ is defined as

$$D = \sqrt{(\max_i x - \min_i x)^2 + (\max_i y - \min_i y)^2}$$

6.2.6 Statistical Disclosure Control

SDC is applied to all indicators besides the ones relating to metadata. K-anonymity can be applied with default value k=20. The value of k is configurable in code before auditing.

6.3 About Acceptance Test

The purpose of the acceptance test is to validate the installation of the Solution in MNO and NSI premises. Acceptance test is out of scope in this PoC project. This test will depend on integration details that are not known in current stage.

7 Annexes

7.1 Algorithm description from Eurostat.

The file *ESTAT 2019.0232 Technical Note - Specification of test use-cases* (v6) is available at https://ec.europa.eu/eurostat/cros/sites/default/files/estat_2019.0232_technical_note_-_specification_of_test_use-cases_0.pdf