D4.1 Initial Status of the Pilot Applications

Document Identification			
Status	Final	**Due Date**	31/03/2019
Version	1.0	**Submission Date**	05/04/2019

Related WP	WP4	**Document Reference**	D4.1
Related Deliverable(s)	D3.1, D6.1, D4.2, D6.2	**Dissemination Level (*)**	PU
Lead Participant	BUL	**Lead Author**	Derek Groen
Contributors	BUL, SZE, PLUS, DIA, ECMWF, MOON, ARH	**Reviewers**	Pawel Wolniewicz (PSNC) Sergiy Gogolenko (HLRS)

Keywords:

Pilot applications, migration, social media, urban pollution, simulation, high performance computing, high performance data analytics.

This document is issued within the frame and for the purpose of the HiDALGO project. This project has received funding from the European Union’s Horizon2020 Framework Programme under Grant Agreement No. 824115. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.

The dissemination of this document reflects only the author’s view and the European Commission is not responsible for any use that may be made of the information it contains. This deliverable is subject to final acceptance by the European Commission.

(*) Dissemination level: **PU**: Public, fully open, e.g. web; **CO:** Confidential, restricted under conditions set out in Model Grant Agreement; **CI:** Classified, **Int =** Internal Working Document, information as referred to in Commission Decision 2001/844/EC.

Document Information

List of Contributors	
Name	Partner
Derek Groen, Simon Taylor	BUL
Stephan Siemen	ECMWF
Zoltan Horvath	SZE
Nabil Ben Said, Amor Messaoud	MOONSTAR
Robert Elsaesser	PLUS
Ruediger Goldschmidt	DIA
Péter Koncz	ARH

Document History			
Version	Date	Change editors	Changes
0.1	01/02/2019	Derek Groen (BUL)	Outline ported to template
0.11	01/02/2019	Derek Groen (BUL)	Instructions for partners added.
0.8	28/02/2019	Derek Groen (BUL)	Merge of a range of separate additions.
0.85	01/03/2019	Derek Groen (BUL)	Addition of missing sections + polishing for consistency and clarity.
0.86	04/03/2019	Derek Groen (BUL)	Added ARH text.
0.87	05/03/2019	Derek Groen (BUL)	Incorporated first round of comments.
0.9	08/03/2019	Derek Groen (BUL)	Incorporated second round of comments.
0.95	20/03/2019	Derek Groen (BUL)	Incorporated internal reviewer comments.
0.99	29/03/2019	Derek Groen (BUL)	Incorporated internal reviewer comments.
1.0	29/03/2019	Derek Groen (BUL)	FINAL VERSION TO BE SUBMITTED

Quality Control		
Role	Who (Partner short name)	Approval Date
Deliverable leader	Derek Groen BUL)	27/03/2019
Quality manager	Marcin Lawenda, PSNC	30/03/2019
Project Coordinator	Francisco Javier Nieto De Santos, ATOS	04/04/2019

Table of Contents

List of Tables

List of Figures

List of Acronyms

Abbreviation / acronym	Description
ABM	Agent-based Model
ACLED	Armed Conflict Locations and Events Database
Dx.y	Deliverable number y belonging to WP x
EC	European Commission
HPC	High Performance Computing
HPDA	High Performance Data Analytics
ML	Machine Learning
UNHCR	United Nations High Commissioner for Refugees
WP	Work Package

Executive Summary

This deliverable reports on the initial status of the pilot applications in HiDALGO. This includes the three core applications (migration, urban air pollution and social media), as well as models and data sources that are planned to be coupled in to these applications. It serves to provide basic awareness of the HiDALGO applications to the consortium and the general public, and helps inform many other activities in HiDALGO, such as the requirements gathering in WP6 and the performance optimization efforts in WP3.

In terms of current status, all three of the pilot applications share a strong science case and impact potential, which is underlined by recognition from external stakeholders such as municipalities and non-governmental organizations such as the UNHCR. The pilots also share a clear definition of their existing workflows, which forms the basis for the technical improvements that are being realized as part of HiDALGO. In terms of performance and scalability, all pilot applications strive to achieve the efficient use of 1,000s of cores by M12, as well as a vast increase in the number of coupled models and incorporated phenomena. Targets for later phases in the project are part of Task 6.1.

However, the pilots differ in priorities in terms of aimed performance and scalability improvements. For the migration pilot, many individual models are available in simplified form, and a major challenge here is to scale up the approach in terms of parallelism, resolution, and range of phenomena incorporated. For the urban air pollution pilot, the main pollution simulation is already in a mature state, but challenges await in incorporating the wide variety of sensor information to make the application more dynamic and suited for optimal use by industry and civic organizations. For the social media pilot, a clear need to analyze the spread of information has been identified, and several key algorithms have been proposed to tackle this problem. Here the development of highly performant and flexible HPDA techniques that can cope with the complex landscape of social media are a key requirement for success. In the case of coupled models, the weather forecasting, telecommunications models, and traffic sensor networks are three priority areas that we will focus on in HiDALGO. Lastly, we foresee the incorporation of other additional models over the course, and have already identified first candidates as part of this deliverable.

1 Introduction

1.1 Purpose of the document

This deliverable (D4.1) is prepared in the context of Work Package 4, which aims to address the definition and implementation of the three pilot applications, making use of the developments of the technical, but also the business-oriented work packages. Specifically, this deliverable contributes to the definition of the three pilot applications, by providing a snapshot of the status of the pilot applications, as of the start of the project (1st December 2018). In addition, this deliverable contributes to the definition of some Pilot case study requirements (as defined in D6.1 – "Requirements Process and Results Definition") and scientific objectives, as well as aims in terms of performance and scalability for the first year.

1.2 Relation to other project work

This deliverable serves as a frame of reference for all application-related research and development in HiDALGO, and reports the state of the applications at the start of the project. It serves to help identify major avenues of impact for WP2, key scalability and performance bottlenecks for WP3, help inform operational activities in WP5, and help identify requirements as part of WP6. This deliverable does not specifically focus on pilot requirements (which are in D6.2) or on providing a roadmap towards coupling and scalability improvements (which will be in D3.2). This deliverable will be largely superseded by D4.2, which is planned to be completed in Month 12.

1.3 Explanation of performance and scalability tables

Most of the document is constructed such that it can be easily interpreted by a wide audience. However, the performance and scalability aspects are somewhat more technical. To facilitate a good understanding of this content, we here provide a brief explanation of these sections, including the corresponding tables.

In this deliverable we seek to indicate the current status of the applications in terms of performance and scalability, as well as the medium-term goal (for month 12). To do so, we provide tables which summarize this information for each of the three pilot applications. For each application, we provide two tables, one for the primary computational kernel, and one for the application as a whole (which includes all coupled models and duplicated instances).

Long term objectives are formulated as part of Task 6.1 (Requirements Analysis and Evaluation) and a technical roadmap will be provided in D3.2.

For the performance and scalability overview for the primary computational kernel, we provide the following information:

Metric	Explanation
Duration of simplest test run	Time it takes for a test run to complete?
Duration of single model production run	Time it takes for a production run to complete, using a typical core count (core count may be given in brackets for extra clarity).
# of agents / elements	Number of agents, particles or unknowns in the model.
Core count per run	How many cores does the code use at most
Max. obtained speedup per run	What is the speedup obtained at that core count?
# of runs used in production	If your main computational kernel is run across a range of configurations in production, how many runs do you typically use?
Data size (input) total	Typical input data size for production runs (all instances aggregated)
Data size (output) total	Typical output data size for production runs.

Table 1: Explanation of the Primary Computation Kernel metrics

For the performance and scalability overview for the application as a whole, we provide the following information:

Metric	Explanation
# of model types coupled in application	How many types of models are coupled in the case study?
Supported coupling mechanisms	How is the coupling data exchanged?
Data size (coupling) total	How much data is exchanged during the coupling activities over the course of a full application run.
# of data source types connected to application	How many types of data sources are coupled in the case study?
Max core count for full application (realistic theoretical estimate)	Aggregating all jobs that reasonably run at the same time, what would the maximum core count reasonably be?

Max core count for full application (achieved)	What is the largest core count you have actually used for your application up to this point?

Table 2: Application as a whole

1.4 Structure of the document

This deliverable outlines the current status of the three pilot applications, and the models and data sources that will be combined with them. The three pilot applications are:

- Migration, where we combine models and data sources to create accurate and scalable simulations for refugee movements.
- Urban air pollution, where we combine a range of sensors and models to model the interplay between traffic flow and air pollution in urban environment.
- Social media, where we combine a range of data sources and social network simulations to investigate and understand the spread of messages on these platforms.

This document is structured in 6 major chapters. We introduce the deliverable in **Chapter 2** and present the status of the pilot applications at the start of HiDALGO respectively for migration (**Chapter 3**), urban pollution (**Chapter 4**) and social media (**Chapter 5**). In **Chapter 6**, we present the models that will be coupled to the pilot applications, and we conclude our deliverable in **Chapter 7**.

2 Initial status of migration case study

There are more than 68 million people forcibly displaced worldwide, of which 24 million are refugees [1].These fleeing individuals are the unfortunate victims of civil wars and internal conflicts, who make decisions to migrate at the times of distress. To understand the causes of forced displacement, researchers establish three concerns faced by migrants, namely, the choice to stay or flee, the choice to flee internally or across borders, and the choice of destination [2]. Their decisions are often based on economic and political push-pull factors in sending and receiving countries. Especially, Schmeidl et al. [3] state that economic and political instabilities, poverty, violence and insecurity in the origin countries push people to flee. In contrary, economically and politically stable and safe countries pull refugees to receiving areas. Thus, we can consider the economic and political conditions, security, the challenges and expenses of moving internally or across borders as causes of forced displacement.

2.1 Science Case

As a baseline for the migration pilot in HiDALGO we have created a simulation development approach (SDA) that allows us to forecast movements of refugees in conflicts. This new approach is important, because it could help to:

- Forecast refugee movements when a conflict erupts, guiding decisions on where to provide food and infrastructure.
- Acquire approximate refugee population estimates in regions where existing data is incomplete, to help prioritize resources to the most important areas.
- Investigate how border closures and other policy decisions are likely to affect the movements and destinations of refugees, to provide policy decision-makers with evidence that could support more effective policy and reduce unintended consequences.

Existing models are largely based on regressing existing refugee data, limiting their predictive power with incomplete or short datasets. Our new approach, of which an early prototype is published already, addresses this weakness by representing refugees as individual agents, combining simple rulesets for individuals to allow large scale and complex movement patterns to emerge. We have already successfully simulated refugee movements in Burundi, Central African Republic, Mali and South Sudan using this prototype, and compared our forecasts with UNHCR refugee camp registrations (see Figure 1 for a validation comparison example). To construct our location graphs (see Figure 2 for an example) and initialize our simulation, we acquire data from UNHCR, the Armed Conflict Location and Events Database (ACLED), and

Bing Maps. With our simulations we can predict the destination of fleeing refugees with approximately 75% accuracy [4].

Although we have been able to construct basic simulations of refugee movements, many major scientific challenges lie ahead of us. For instance, we wish to improve the accuracy and resolution of our model such that we not only can make predictions for the destinations of refugee movements, but also reproduce the routes that they are likely to take in their travel. This would allow NGOs and policy makers to have more accurate forecasts on the travel duration of refugees, and could help efforts to choose optimal camp locations. This will inevitably not only lead to an increase in locations and roads (a natural consequence of using more fine-grained location graphs), but also lead to more sophisticated decision models for the agents in the simulation.

In addition, we seek to clarify the role of weather and telecommunication in our simulations. The former could affect the journeys that refugees are likely to undertake, while the latter may help us identify the typical routes that refugees take in their journey towards their destination.

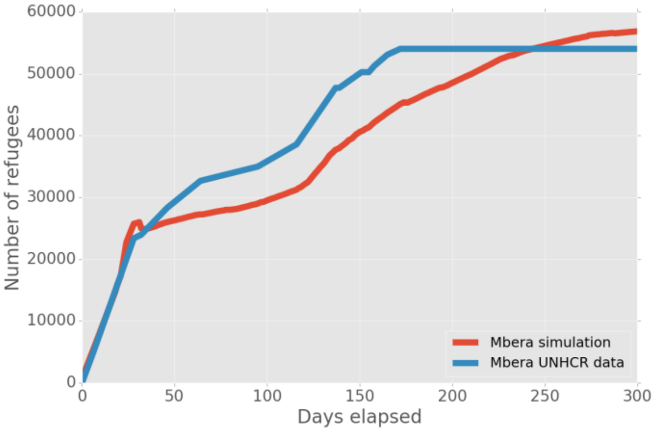


Figure 1 - Example comparison of forecast predictions and UNHCR data (for the Mbera camp in the Mali conflict).

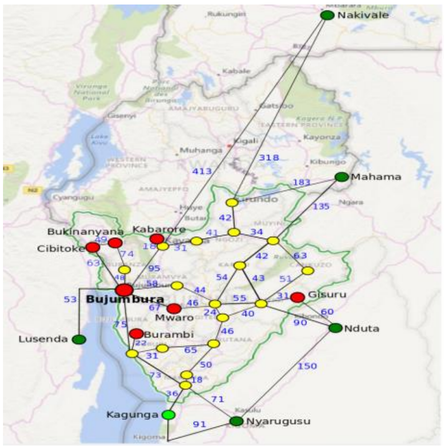


Figure 2 - Example network used in FLEE (from the Burundi conflict simulation).

2.2 Algorithmic Overview

Our simulations are part of a Simulation Development Approach, which is discussed in detail in [4] and summarized here. Here, we first select a country and time period of a specific conflict, and second we obtain relevant data to the conflict from the three data sources: ACLED (http://www.acleddata.com/), the UNHCR database (http://data2.unhcr.org/), and the Bing Maps platform (https://maps.bing.com). We use ACLED to obtain the locations and dates of battles that have taken place in the conflict, and the UNHCR database to obtain the number of refugees in the conflict, as well as the camp locations and capacities. We rely on the Bing Maps platform to obtain locations of major settlements and routing information between the various camps, conflict zones and other settlements.

In the third phase, we construct our initial simulation model using these data sets, and create among other things a network-based ABM model. Once we have constructed the initial mode, we refine it as part of the fourth phase. Here, we manually extract population data to help determine where refugees flee from, as well as information on border closures and forced redirections of refugees.

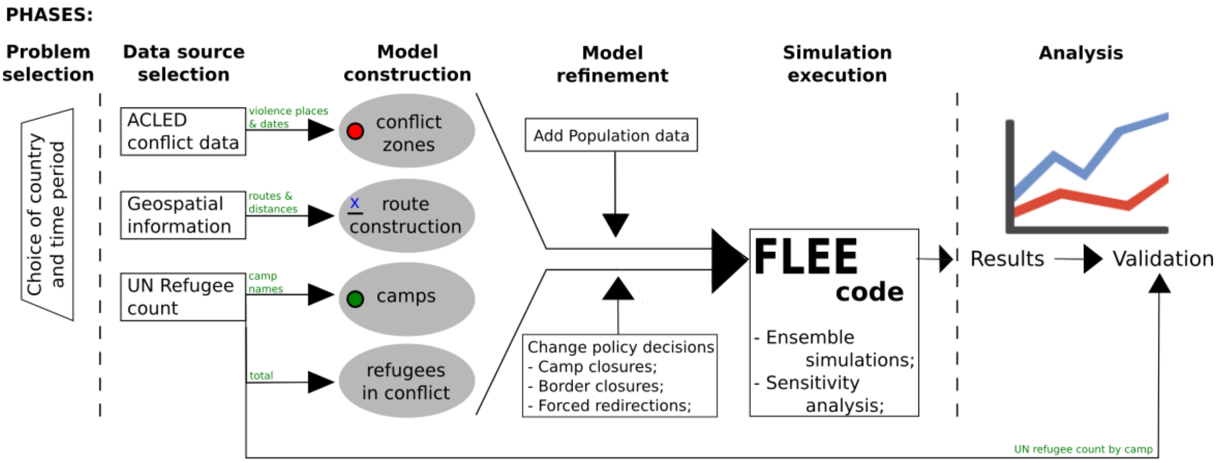


Figure 3 - Simulation Development Approach used for agent-based simulations of refugee movements. This is an evolved version from the one presented in [4].

2.2.1 Main model Overview

We currently run our simulations using the serial Python-based FLEE code, which is available at http://www.github.com/djgroen/flee-release.

Flee relies on an agent-based modelling approach with a location network graph, where each refugee in the simulation is represented by a single agent. The location network graph contains four types of locations (vertices): conflict zones (refugee agents depart here), camps (agents tend to finish their journeys here), regular locations (agents may stay or move elsewhere) and forced redirection points (agents will be forwarded along a fixed route, representing government-orchestrated movements). Each location features the following main attributes:

- Civil population, or capacity (in the case of camps)
- # of agents residing in location
- GPS coordinates
- Attraction scores, which are recalculated dynamically at each time step.
- List of travel routes (edges)

The edges in the graphs represent travel routes, which feature as attributes a length in km, # of agents on the route, and the two end points. In most cases these edges represent roads interconnecting locations, with their lengths calculated using the Bing Maps route planner. However, in countries with very few roads (e.g., South Sudan) we also represent key walking routes as edges. Walking routes are represented as edges with a modified weighting, to reflect their reduced accessibility. Within Flee, agents are represented in a very basic fashion, and more features are likely to be added during the project. As attributes they currently have their current location, home location, and timesteps elapsed since departure.

Flee has been parallelized using a basic parallelization algorithm [5], where agents are uniformly distributed across the processes.

The main algorithm (`evolve()`) which propagates the system by one time step is currently structured as follows:

1. Update location scores (which determine the attractiveness of locations to agents).
2. Evolve all agents on local process.
3. Aggregate agent totals across processes.
4. Complete the travel, for agents that have not done so already.
5. Aggregate agent totals across processes.
6. Increment simulated time counter.

One requires two `MPI_AllGather()` operations per iteration loop, during steps 3 and 5, but this will increase as the ruleset becomes more sophisticated. Our existing refugee simulations currently require 300–1000 iterations per simulation, which would result in 600–2000 AllGather operations. These operations require all processes to synchronize. Flee also supports coupling communications with other codes and is able to exchange refugee agent objects with other codes operating at smaller or larger spatiotemporal scales [5].

We have combined FLEE with the FabFlee automation toolkit, which allows us to easily explore different policy decisions and perform sensitivity analysis across parameters using remote supercomputers. As a result, we are currently able to run large sets of serial simulations efficiently, using a basic high-throughput style approach.

2.2.1 Data types and workflow

An example workflow for a single run of the main model is provided in the Figure below:

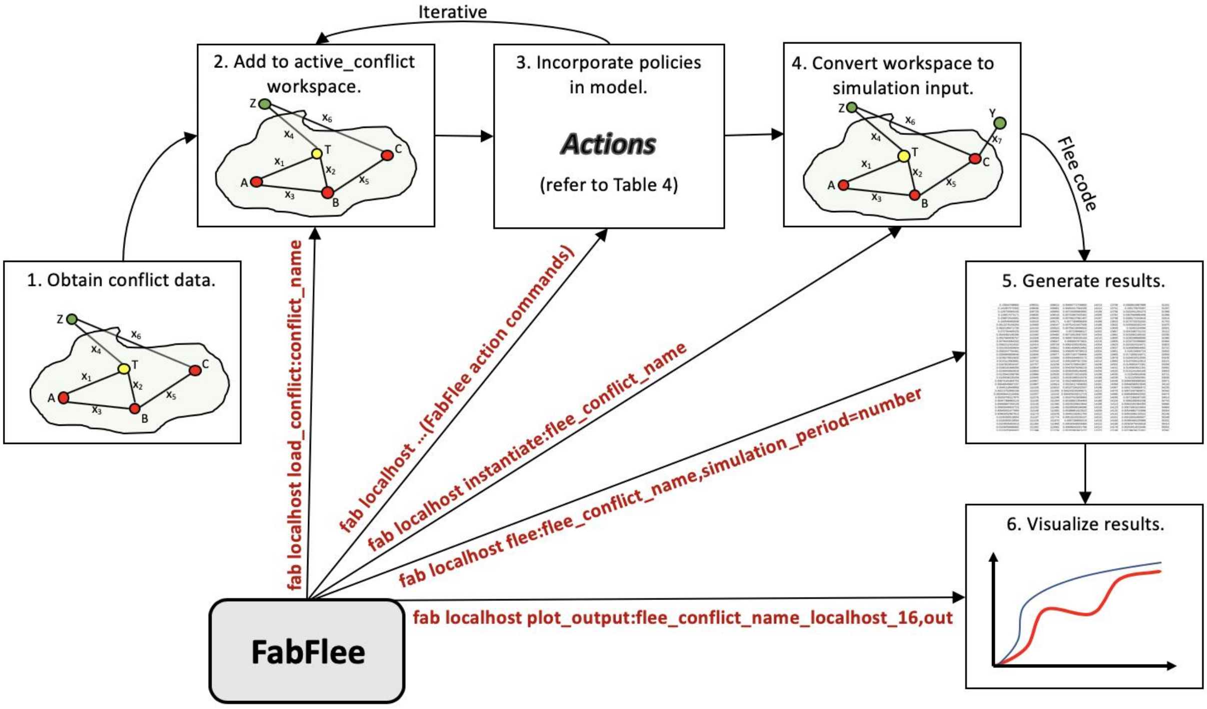

Iterative
1. Obtain conflict data.
2. Add to active_conflict workspace.
3. Incorporate policies in model.
Actions
(refer to Table 4)
4. Convert workspace to simulation input.
Flee code
5. Generate results.
6. Visualize results.
Z
X6
X4
T
C
X1
X2
X5
A
X3
B
Y
X7
FabFlee
fab localhost load_conflict:conflict_name
fab localhost ...(FabFlee action commands)
fab localhost instantiate:flee_conflict_name
fab localhost flee:flee_conflict_name,simulation_period=number
fab localhost plot_output:flee_conflict_name_localhost_16,out

Figure 4 - Example workfow where a refugee simulation is created, refined, executed and analyzed. Courtesy of Suleimenova et al. (in prep.).

A full-blown refugee movement simulation typically consists of at least 100 of these runs, as the code needs to be tested for sensitivity against a range of key parameters. In addition, the simulation must undergo an extra iteration of execution and sensitivity analysis for each policy decision or counterfactual scenario that has been incorporated. Examples of such scenarios include runs with border closures, or runs that have extra camps added, or some of the existing camps removed.

Both input and output are currently supplied as CSV files, and the size of these files is currently in the order of megabytes per run. However, the output data contains only total counts of refugees at the camps, and we expect the file size and the number of files to increase as we add additional detail to the output (e.g., information on the journeys of individual refugees. We will work closely with WP3 and WP6 to address any performance challenges arising from this growth in I/O.

2.3 Goals for HiDALGO

The main scientific and impact goals for the migration pilot, over the course of HiDALGO are to:

- Increase the level of detail in the model, for instance by adopting a more fine-grained location graph.
- Incorporate a broader range of relevant phenomena, such as weather conditions, communications, and food security, through model coupling.
- Validate against a broader range of historical conflicts, and a broader range of data sources (for instance, validate refugee journeys against telecommunications data).
- Improve the scalability and speed of the application, and reduce the time required to construct simulations.
- Establish improved uptake of the code with governments, NGOs, and the wider academic community.
- Extend the range of applicability for the simulation approach, for instance by supporting the modelling of internally displaced people.

2.3.1 Performance and Scalability

Although the current implementation of Flee has been validated against observational data, it is still in a prototype stage regarding the parallelization. We therefore expect to make substantial scalability improvements for the main model in the first 12 months of the project, and scale beyond the single node in year 2 and 3, as the simulations will then contain more detailed location graphs and more sophisticated rule sets. In addition, we aim to eventually couple 6 different model types in this pilot. However, each new model type will be subject to careful review, and may be left out in production runs if it does not yet provide a meaningful improvement in validation tests. We expect the core count used by coupled models to increase heavily in year 2 and 3, as we link up to production weather simulations, and will likely require ensemble of multiple instances for some of the other coupled models (e.g., conflict propagation and microscale models).

Metric	M0 value	M12 goal
Duration of simplest test run	10 s	10 s
Duration of single model production run	3 h	3 h
# of agents / elements	300,000	2,000,000

Metric	M0 value	M12 goal
Core count per run	24	32
Max. obtained speedup per run	8	24
# of runs used in production	120	3000
Data size (input) total	1 MB	100 MB
Data size (output) total	20 MB	4 GB

Table 3: Primary Computation Kernel

Metric	M0 value	M12 goal
# of model types coupled in application	2 (main ABM, and microscale ABM)	4 (+ conflict, weather model)
Supported coupling mechanisms	File-based only	File-based + TCP-based
Data size (coupling) total	1MB	100MB
# of data source types connected to application	3 (ACLED, UNHCR, Bing maps)	5
Max core count for full application (realistic theoretical estimate)	24 (2 cores per job, running 12 jobs in parallel)	7,000 (300 jobs * 24 cores + 200 cores for coupled models)
Max core count for full application (achieved)	4 (no production on HPC yet, only testing)	n/a

Table 4: Application as a whole

2.4 Evidence of External Recognition

The FLEE code has been published in several peer-reviewed articles, including a Nature Scientific Reports paper and a Winter Simulation Conference paper. It has also been a major target application in two grant applications that have been accepted. These include HiDALGO of course, as well as the Verified Exascale Computing for Multiscale Applications Horizon 2020 grant, (VECMA, http://www.vecma.eu). The FLEE code has been installed and tested by researchers from UNHCR shortly after the Scientific Reports paper was published, and the work led us to becoming a partner the Search and Rescue Observatory in the Mediterranean (SAROBMED, http://www.sarobmed.org). The work also led to invited talks at the University of Amsterdam and the University of Geneva, and has been nominated to become an Impact Case Study for Brunel University London for the 2021 Research Excellence Framework. Lastly, the work led to the granting of a Brunel-funded Impact Accelerator Award of £21k, which will be used in part to disseminate our efforts across different communities in Africa.

3 Initial status of urban air pollution case study

Urban citizens suffer more and more from bad air quality in many cities resulting plenty of premature deaths, as media has reported recently (see e.g. [6],[7],[8]). One of the most severe pollutant is NO_2 of which main producer is the vehicular traffic. To reduce air pollution urban traffic will need to be controlled (see [9]). The vision of HiDALGO's urban air pollution application is to make cleaner air in cities using high performance computing, sensor data acquisition and data analytics.

To achieve this, the pilot will develop a computational tool as a service to policy makers and the civil society, that accurately and quickly forecasts air pollution at urban street level. The pilot will also develop a traffic control system to minimize air pollution while considering traffic flow constraints. This service will couple real time traffic and air quality sensor data, delivered by MK and ARH, and meteorological prediction data, delivered by ECMWF, to an urban air flow system which is to be simulated with HiDALGO's HPC and HPDA tools, using the latest mathematical algorithms. In doing so, the pilot will establish a digital twin for urban air pollution.

The initial phase of the pilot is established by adapting the MSO4SC project's 3DAirQualityPrediction application, which is a coupled HPC simulation of urban traffic and multicomponent air flow. During the HiDALGO-project, SZE will further reshape and extend this application to achieve the goals stated above. The developed system will be demonstrated to model pollution in the city of Győr, Hungary.

3.1 Science Case

Model-based digital twins are considered to be a top technology trend by the main business analysts and leaders of industry (see e.g., [10],[11] and their references). However, construction of a model based digital twin of a physical asset, as in our case for the urban air pollution pilot, is challenging due to the complexity of coupling HPC simulation, data assimilation from several sources, data acquisition and HPDA, all of which require research activities. We summarize the main research goals of the urban air pollution pilot application in the table below.

Nr.	Research goals	Challenges
1.	Accurate and fast CFD simulation of the multicomponent air flow in cities.	1. Large scale (5-10 km horizontally and 0.5 km vertically) and complexity (ground, buildings, parks) of the geometry.

Nr.	Research goals	Challenges
		2. Accurate result is needed at street level, i.e. cell size of the mesh should be maximum 5-10 meter horizontally near the ground. 3. Multiphysics nature of the physical process: nonlinear transport of the air; advection, diffusion, reaction and absorption of the pollutant species; heat radiation; turbulence caused by traffic; etc. to be modelled for accuracy to a reasonable extent. 4. Assimilation of meteorological forecast data for boundary conditions of the air flow. 5. HPC scalability of the multicomponent codes.
2.	Model order reduction of the air flow simulation.	Model reduction techniques, e.g. combination of POD variants and HPDA methods (e.g. clustering) proved very efficient for dissipative systems [21]. In our pilot we must account for strongly advective (i.e. high wind speed) cases as well, which is still under research by the community.
3.	Accurate and fast simulation of the urban traffic using real time traffic sensor data.	Assimilate sensor data from traffic real time and build fast traffic simulation based upon that.
4.	Build and operate the sensor network for accurate traffic sensing.	Accurate sensing, i.e. provide accurate origin-destination information via recording plate numbers of vehicles.
5.	Compute emissions from all major sources.	For NO_x pollution, traffic is the main source. We should incorporate more sources to be able to consider $PM_{10}/PM_{2.5}$: emission from industry and buildings; boundary sources and long transmission sources.
6.	Coupling of the different algorithmic components.	Though weak coupling is assumed to be applied first, coupling is a challenge due to many components.
7.	Optimization of the traffic: algorithm and traffic operations.	We require multiobjective optimization for traffic and air quality objectives with discrete design (traffic light programs) under CFD constraints for real time operations. This is highly challenging as it constitutes an optimal control problem. We aim to obtain an approximate solution by using model predictive control (MPC).

Table 5: Urban air pilot main research goals

To achieve the research goals the pilot will build on the theory and technology of the state-of-the-art of HPC, HPDA and mathematical algorithms. During the project, the primary aim is to provide a production-ready service which is robust, fast, scalable and inexpensive, with sufficient accuracy to pass all the necessary validation tests.

3.2 Algorithmic Overview

We present the work flow with intermediate data types in Figure 5. This work flow describes the initial state of the pilot, i.e. the scientific challenges 1.1-3 in Section 4.1. The algorithms of the further phases correspond to the other scientific goals of Section 4.1 and will be expanded upon in future deliverables.

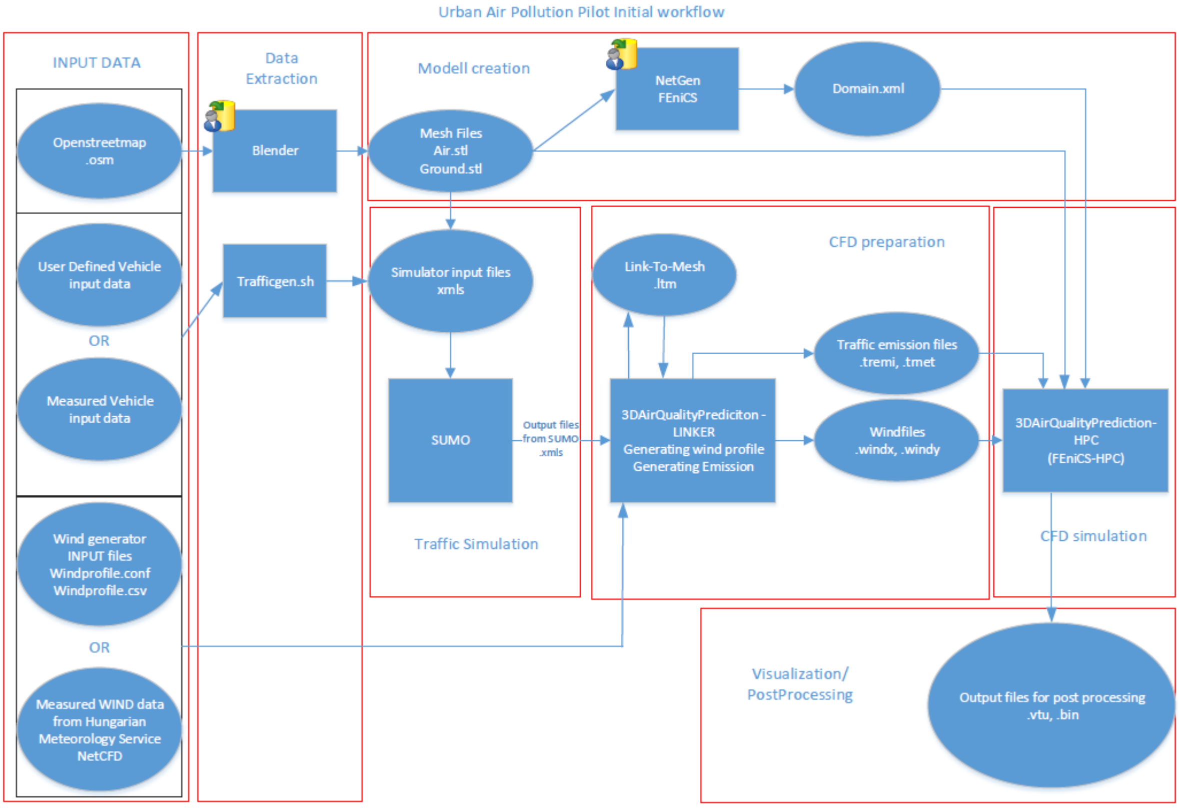

Urban Air Pollution Pilot Initial workflow
INPUT DATA
Openstreetmap .osm
User Defined Vehicle input data
OR
Measured Vehicle input data
Wind generator INPUT files Windprofile.conf Windprofile.csv
OR
Measured WIND data from Hungarian Meteorology Service NetCFD
Data Extraction
Blender
Trafficgen.sh
Modell creation
Mesh Files Air.stl Ground.stl
NetGen FEniCS
Domain.xml
Simulator input files xmls
SUMO
Output files from SUMO .xmls
Traffic Simulation
CFD preparation
Link-To-Mesh .ltm
3DAirQualityPrediciton - LINKER Generating wind profile Generating Emission
Traffic emission files .tremi, .tmet
Windfiles .windx, .windy
3DAirQualityPrediction-HPC (FEniCS-HPC)
CFD simulation
Visualization/ PostProcessing
Output files for post processing .vtu, .bin

Figure 5 - Workflow overview of the urban pollution application.

3.3 Goals for HiDALGO

Over the course of HiDALGO we would like to achieve the research goals written in details in Section 4.1. These all will enable the project to define a service to be provided to stakeholders specified in detail in Section 4.4.

3.3.1 Performance and Scalability

For the urban pollution pilot, we mainly strive to achieve performance and scalability improvements by incorporating additional models in the application. We already have access to several mature simulation codes for the primary computational kernel, and as such do not focus on optimizing this kernel during the first year. For M12, we strive to integrate four model types in the application and be able to scale the full application to approximately 64,000 cores, due to the presence of these additional models.

Metric	M0 value	M12 goal
Duration of simplest test run (~2000s simulation time)	~2 hours	~2 hours
Duration of single model production run (~84000s simulation time)	~3 days	~3 days
#elements	3.000.000	3.000.000
Core count per run	32 cores	32 cores
Max. obtained speedup per run	10	10
# of runs used in production	1 (in general, 70% of runs are done in production)	1 (in general, 70% of runs are done in production)
Data size (input) total	~10MB	~10MB
Data size (output) total	120GB	200GB

Table 6: Primary Computation Kernel

Metric	M0 value	M12 goal
# of model types coupled in application	2	4
Supported coupling mechanisms	weak, offline	weak, offline and real time
Data size (coupling) total	10MB	10MB+simulation time*(10MB/hours +0.1MB/minute)
# of data source types connected to application	0	1
Max core count for full application (realistic theoretical estimate)	128	64000
Max core count for full application (achieved)	64	N/A

Table 7: Application as a whole

3.4 Achieving External Recognition

Existing and potential stakeholders of the urban air pollution application cover a wide range of business sectors and science communities. We summarize them in Table 8.

Stakeholder	Interest in the urban air pollution application
City authorities (Győr), government units responsible for air pollution (Hungary)	Analysis of different planning scenarios.
Hungarian environmental agencies and healthcare	Better insight to spatial resolution of urban air quality indicators and opportunity of studying.
City habitants (Győr)	Available online tool for checking the status and prediction of air pollution at in the city, with high resolution (at street level). Requirement: the city (like Győr, the demonstration city of the HiDALGO-project) provides accurate emission data.
Environmental scientists	Test bed for trying own models in traffic, emission, dispersion modules.
Traffic authorities	Control system to optimize traffic and immission due to it.
Automotive industry	Analysis of the effect of vehicle emission scenarios.
Computer scientists, HPC experts	Benchmark low level HPC solutions and infrastructure on this very large scale computation.
Data scientists	Test different HPDA methods on a data rich infrastructure with real time data acquisition.
Mathematicians	Benchmark environment of different algorithms for model reduction.
Transport researchers	The HiDALGO infrastructure provides sensor data for very sophisticated traffic modelling with its car plate recording and actuating systems.
Digital twin developers	The application provides a workflow for general digital twins to be developed on the HiDALGO infrastructure.

Table 8 - Summary of key stakeholders for the urban pollution application.

4 Initial status of social networks case study

The main goal of this case study is to understand, model and simulate the spread of messages in various social networks. To perform this task, several scientific aspects need to be clarified: the structural properties of social networks, the stochastic behaviour of information spreading. In addition, several programming related problems need to be addressed, such as the deployment and efficient simulation of the aforementioned process on parallel machines. Solving these problems will allow us to identify the properties of so-called malicious messages and to propose countermeasures to point out such messages. To achieve the main objective, we divide our task into several parts, each addressing different problems arising within this pilot application.

4.1 Science Case

Social networks are omnipresent in our daily life and influence our behaviour. Political decision makers as well as most economic players use social media channels to reach out and attract the attention of most individuals. Furthermore, many people of different age use social networks to interact with each other. Understanding the spread of messages in these networks will help these decision makers. The main goal of this pilot is to accurately model and simulate the spread of messages in social networks on a large scale. To achieve this goal, we pursue the following tasks:

1. Analyze the structural properties of social networks.
2. Construct synthetic graph models for these networks and develop a validation framework for them.
3. Analyze the stochastic behaviour of the messages spreading among the users and perform accompanying analyses for them.
4. Combine the synthetic models developed for social networks and the spread of messages.
5. Build a simulation framework upon these models.

At the beginning, all these tasks are addressed separately. For the first two tasks, we already have existing results.

First, to understand the structural properties of social networks, we analyse different types of networks, which can either be accessed through Stanford Large Network Dataset Collection in anonymized form or are obtained through crawling (e.g. Twitter). Here several ethics issues have to be taken into account, which we discuss in Deliverable 8.1. Apart from well known characteristics, such as the degree distribution, diameter, distance distribution, or the clustering coefficient, we aim to understand the size and structure of the communities in the underlying graphs. For this, we use clustering to obtain local partitions of the neighbourhoods of nodes and sophisticated merging techniques to get an overlapping clustering of the network (see next subsection). The distribution of the cluster sizes, the number of clusters assigned to the nodes, and the distribution of edges within clusters constitute the foundation of the synthetic graph models to be developed.

Second, the synthetic graph models we build heavily rely on the properties described above. To construct these graphs, we apply a random graph approach, where the nodes and clusters follow the distributions obtained in task 1. Several building blocks of the graph building algorithm are yet to be determined. The goal is that the resulting random graphs have similar properties as the original networks. For the validation, we compute the distribution of the eigenvalues of the two graphs (original network vs. synthetic graph), and test whether the two follow the same distribution up to some small deviation (see next subsection for the algorithm). In the case of certain networks (e.g. Twitter) we are going to access only parts of these networks.

There are several open questions w.r.t. the last three points. In task 3, we have to analyse the probability distribution of the spread of a message from one user to the next. Here, various structural aspects have to be determined, which influence this probability distribution. In the combined model (task 4), we have to take into account scalability and efficiency. In task 5, an agent-based model has to be built on the top of the combined model. Here, the main challenges are to design a framework, which encompasses node level performance, parallel scalability and algorithmic efficiency.

DIALOGIK (DIA) supports and supplements the research activities of the case study. A basic assumption of this research can be that content related factors influence the information spread. We currently elaborate our research strategy based on an unsystematic review of publications and the interactive exchange with the relevant partners. Considering the numerous research approaches on information diffusion, two basic methodologies are promising, which can be implemented alone, simultaneously (interaction of research methods) or sequentially (combined as pre and main study):

- **Qualitative interviews**: Interviews with relevant stakeholders can provide a kind of community and peer feedback during the initial research phase of the pilot study or the main phase (one shot) or both. Qualitative interviews allow, for instance, exploring the perspectives of researchers and practitioners in the field according to the topics of information diffusion in social networks, hereby specifically the characteristics and distribution of malicious messages, as well as their influence on the information diffusion of Twitter messages (e.g. acceleration, deceleration and suppression of diffusion effects). Qualitative interviews, firstly, can indicate relevant substantial aspects for refining the simulation models (model specification). Secondly, interviews can contribute to the validity of the simulation models or the underlying conceptual and empirical research work including the data sampling and analysis procedures. The interviews, in addition, can support other work packages, for instance, to identify stakeholders interested in the participation as external partner or as user of the HiDALGO services.

- **Quantitative analyses**: An internal strategy paper, summarising the options for the quantitative analyses, will be available mid of April 2019. For a better understanding, we describe some of these options in the following. The first two options can be implemented manually (very small samples) or by analysis software (small samples). They represent a mixture of quantitative and qualitative (interpretive) methods and are 'explorative'. A sample can be extracted from a bigger data volume of Twitter messages. Messages of this sample could be characterised and typologised based on conceptually (pre-defined) or empirically derived criteria. The frequencies of individual features or the frequency of certain combinations of features observed in the real world networks could be explored. Another option is to compare messages from pre-characterised sources based on certain criteria. For instance, the information spread activity of Twitter sources with an observed focus, including fake news, can be compared with sources without such a focus. A meaningful additional option is to implement a longitudinal design with several analysis waves. A machine learning approach (mere quantitative analysis) allows the analysis of bigger message samples and could be used during the runtime of the project. The relevance of these methodical alternatives will be weighted considering the overall research strategy of DIA as well as the given regulation on the protection of data privacy.

4.2 Algorithmic Overview

As described above, the work in this pilot can be divided into five major parts. Here we give the main workflow (see Figure 6 and Deliverable 6.2), and describe the algorithms used in the first two parts so far. For the parts 3.-5., the algorithms and methods still have to be developed. We use clustering to understand the community structure of these networks. First, the neighbourhood of each node v is clustered using the parallelized Louvain clustering algorithm [12][13] from the NetworkKit tool suite [14], and then v is assigned to each of these clusters. By this method, we obtain a huge number of overlapping clusters. Then, clusters contained in other clusters are deleted and we apply a merging procedure to get a consolidated clustering. Here, we merge two (partially) overlapping clusters, if the second smallest eigenvalue of the obtained cluster is higher than the ones of the two clusters to be merged. The merging is performed in a greedy way so that the resulting clustering ends in a local optimum w.r.t. the second smallest eigenvalues of the obtained clusters.

To compare our synthetic models with the original network, we compute the eigenvalue distribution of the normalized Laplacian of the corresponding graphs. Even though many libraries for computing eigenvalues exist (see e.g. [15]), their performance depends heavily on the number of eigenvalues that are requested. Usually, only a small part of the spectrum needs to be computed and the eigenvalues do not lie dense next to each other. In our case, most of the values lie tightly concentrated around 1, leading to a poor performance. This, combined with the fact that we only need to compute a histogram of the spectrum (and not necessarily the spectrum itself), leads to another approach. In this modified approach, we use the fact that all eigenvalues of the normalized Laplacian are in the range [0,2]. In order to obtain the histogram mentioned above, we compute the number of eigenvalues in disjoint ranges of predefined length covering the whole interval [0,2] (currently, the length of one range is 0.02 leading to 100 ranges in total). For this, we use a modified version of the algorithm in [16].

Our software is implemented in C++. We use the SLEPc [15] and PETSc [17] libraries, which provide us with an interface to the MUMPS [18] library. All the above are designed to work on large scale distributed systems. The first two allow us to perform basic algebraic operations, such as matrix-vector subtraction, and also a rigorous computation of all eigenvalues of our matrix, which is not feasible in our case. Finally, MUMPS implements the distributed LDL-decomposition. As a pre-processing step, before factorization, the PARMETIS library [19] is used as well.

The computation of a single range is designed to run on a parallel machine. Furthermore, the computations of the different ranges are independent from each other, so we can achieve high scalability when computing the complete histogram.

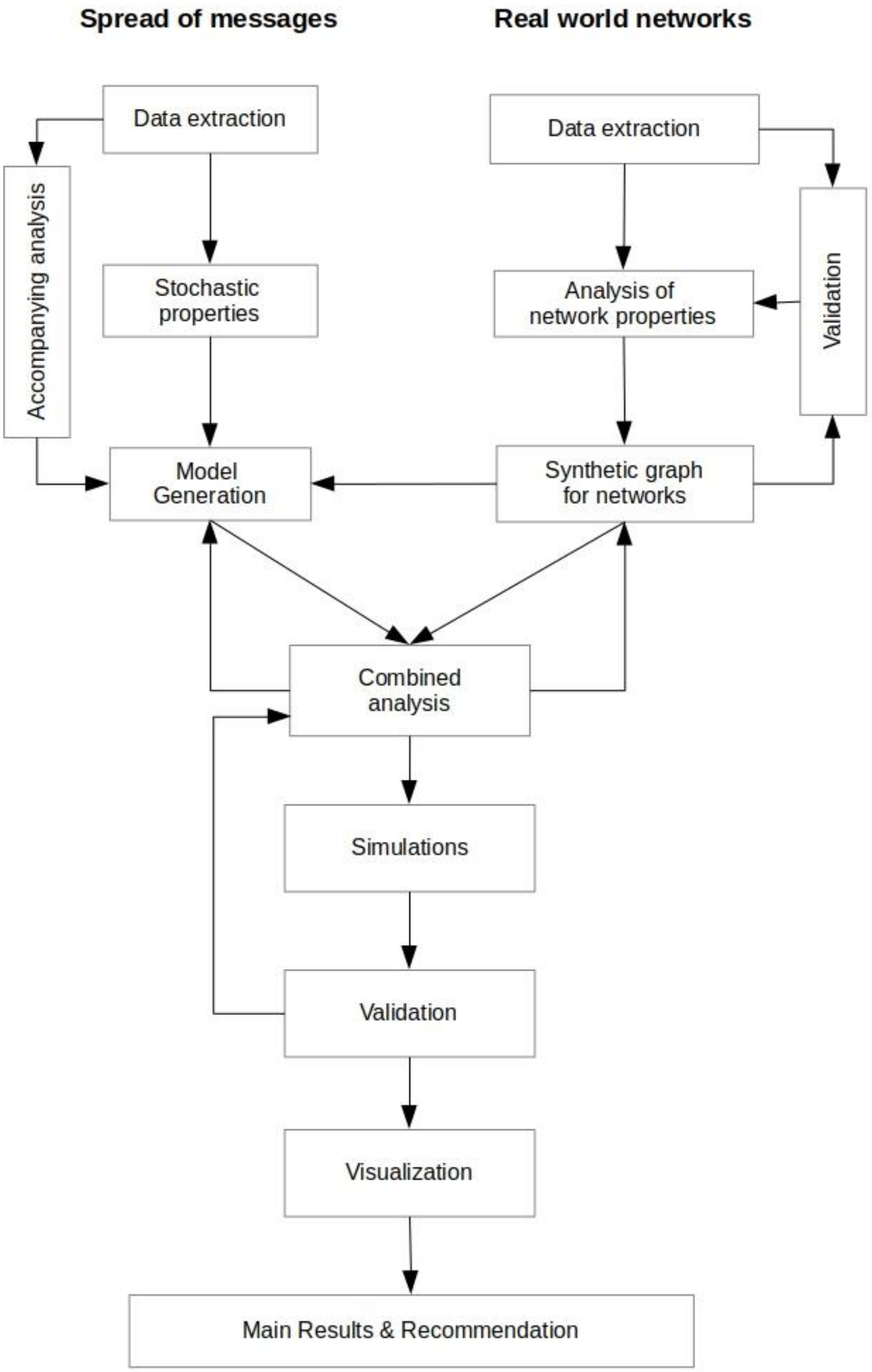


Figure 6 - Workflow of the social networks pilot.

4.3 Goals for HiDALGO

The main goal of this pilot is to have a functional framework for simulating the "life" of social networks. To achieve this goal, we defined the following milestones. During the project's lifetime, these milestones will be adjusted to accommodate our theoretical findings.

4.3.1 Performance and Scalability

In the case of the social networks pilot, many of the key analyses and modelling algorithms are not effective a priori, but will be established as part of the activities in HiDALGO. Also, the approach in this use case differs at several places from that of the previous use cases, e.g. we will not deploy simple simulation runs over the spread of messages in social networks. Instead, we work first on the five tasks defined in Sec. 5.1, and run the simulations after all these blocks have been optimised. Also, it would be premature to give any speedup estimates at this stage. To reflect all these, we present a modified performance and scalability overview, emphasizing the targets that are particularly essential to the development of this application.

The estimates given for M12 take into account the scalability of our code observed so far, as well as the increase in the input size and possible future optimizations of the code. Clearly, the final results may deviate from the estimates we give in this table.

Metric	M0 value	M12 goal
Duration / # cores for clustering	14.000 s /100 cores	35.000 s / 2000 cores
Duration / # cores for eigenvalue distribution for one range	8000 s / 2760 cores	35.000 s / 5520 cores
Duration / # cores of one message simulation	n/a	Up to 60 s / 24 cores
Duration / core count per overall simulation run	n/a	Up to 120.000 s / 25.000 cores
Size of the input – overall simulation	n/a	Network with 1 Mio nodes and 5000 messages / up to 3 GB

Table 9: Primary Computation Kernel

The networks we are going to cluster and evaluate w.r.t. their eigenvalue distribution by M12 are significantly larger than the ones we analyse now (they differ roughly by a factor of 5).

Concerning the results w.r.t. one message simulation, we are going to utilize one compute node of Hazel Hen with 24 cores. When we simulate up to 5,000 messages, one compute node will be used to simulate the spread of several messages (between 5 and 100, depending on the message). The dependencies between messages must also be taken into account, which may lead to high communication complexity between the allocated compute nodes.

Metric	M0 value	M12 goal
Supported coupling mechanisms	n/a	File based / Open MP / MPI
# of data source types connected to application	n/a	2
Max core count for full application	n/a	25.000

Table 10: Application as a whole

4.4 Achieving External Recognition

This implementation of the building blocks of this pilot application has started a few months ago and thus, the potential stakeholders have still to be identified. At this stage it would be premature to talk about existing external recognition. At first place, we plan to attract the attention of two main stakeholder groups: the scientific community working in the area of real world networks / information spreading and the decision makers interested in the impact of social media on the society. Concerning the first group, in this pilot we seek to address several important research questions and will develop a range of computational approaches to do so. Concerning the second group, we envision that the simulation results will aid decision-making for industrial players and other managerial roles. By understanding and simulating the spread of messages in social networks we may be able to identify malicious messages, which intend to influence the behaviour of a large number of people. Such false messages have been observed for example during the US presidential election campaign in 2016 or before the Brexit referendum. We argue that systematic identification of such messages could benefit society at large. The dissemination strategy for our results in this pilot is work in progress and will be addressed in deliverable 7.2.

5 Initial status of coupling applications and data sources

Very seldom models or simulations are run in isolation. In reality, many services are built as a chain of models/simulations combined with different post-processing steps triggered as follow-up steps. With the emergence of Machine Learning it has become even more attractive to build models based on past observations.

In recent years there has also been increasing efforts to bring together data sources from different domains to increase the value of services. Traditionally domains have their own data formats, terminology and conventions which in practice meant it was not always easy to combine data easily.

This section describes the status the various coupling applications and data sources that we seek to introduce in the HiDALGO pilots, as of at the start of the project. The coupling work will be strongly driven by the case studies described above, and M12 aims in regards to coupling are described there, while detailed requirements and roadmaps are part of the WP6 and WP3 deliverables. In particular D3.2 will contain a comprehensive coupling roadmap, as at that time the requirements from all applications have been incorporated in the project work. In general, we aim for simple file-based coupling in the first year, and a much closer coupling which also allows feedback between systems after M12.

5.1 Weather Simulations

Will couple with: Migration and Urban Pollution Pilots

Weather forecasting as a domain has for many years practiced the coupling of models and post-processing tasks. It has been shown, that coupling of ocean and wave models to atmospheric models can have significant improvements to the medium-range and seasonal forecasts. High resolution Limited Area Models (LAMs) for forecasts of up to 3 days are only possible because they are coupled to global lesser resolution models to provide them with the bounding conditions to their domain. Various post-processing tasks to detect extreme weather events, such as hurricanes, are triggered from these model runs. All this work is highly time critical, since a weather forecast has to be issued in a very short time windows to be of use.

ECMWF with its medium-range global model is in a very central role to many of these couplings in this domain. Indeed, many National Weather Services (NWS) and international co-operations, like Copernicus programme of the European Commission, make use of ECMWF models. The ECMWF Integrated Forecasting System (IFS) consists of several components coupled together in different ways:

- an atmospheric model run at various resolutions appropriate to the forecast length (high resolution (HRES), ensemble (ENS), extended-range, and seasonal forecast).
- an ocean wave model (ECWAM) run with various configurations (HRES-WAM, HREW-SAW, ...)
- an ocean model (NEMO) including a sea ice model, the Louvain-la-Neuve Sea Ice Model (LIM2).
- a land surface model (HTESSEL) including a lake model (FLake).
- a data analysis system (4D-VAR).
- perturbation techniques for generation of the ensembles.

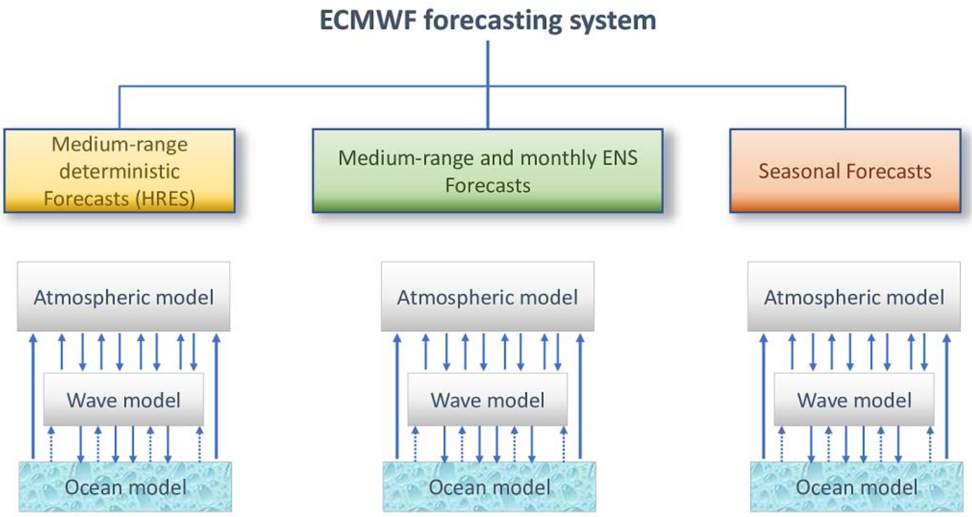


Figure 7 - ECMWF Integrated Forecasting System (IFS) Illustrates interactions between components of the IFS.

Currently the couplings are either very close running on the HPC or loosely coupled by one model writing out its full output and other models/services triggered from it.

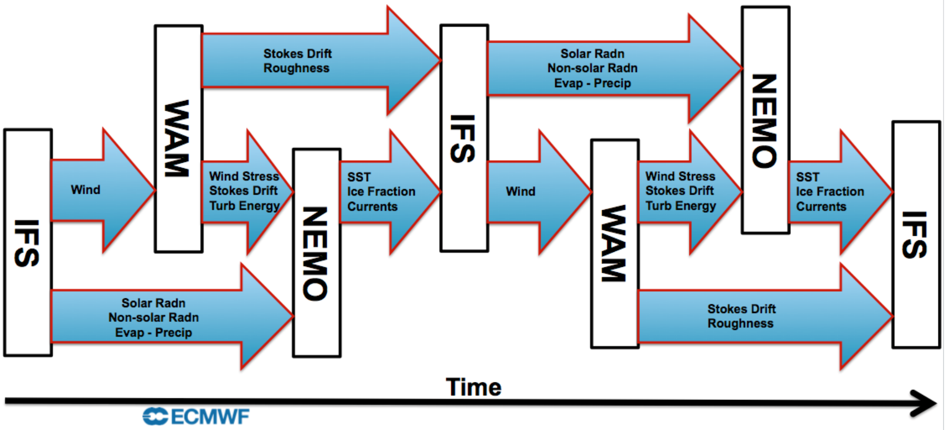
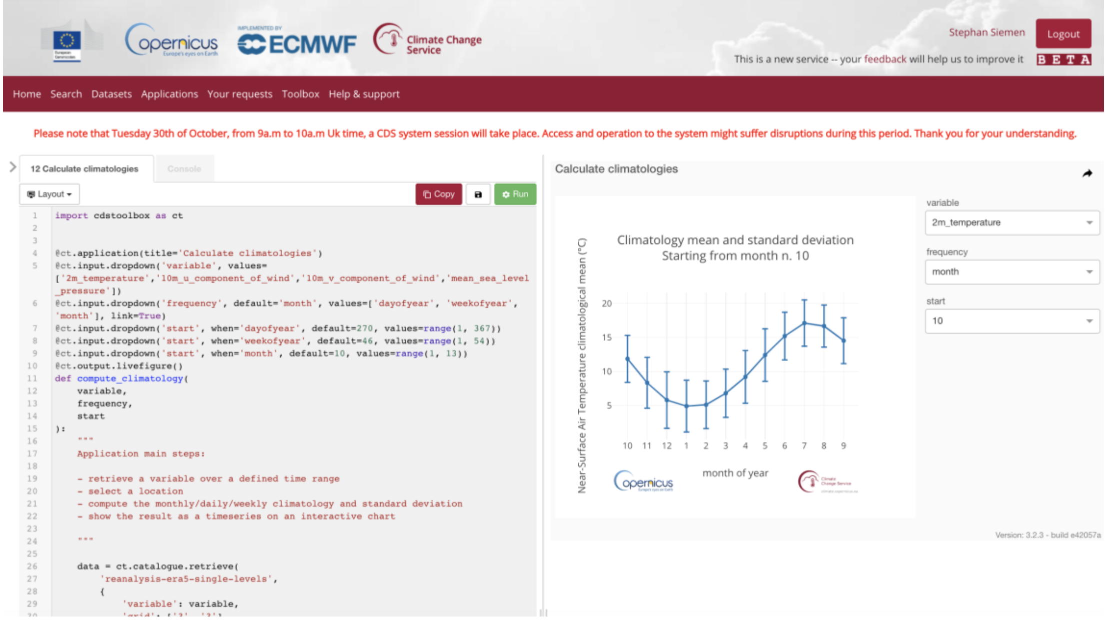


Figure 8 - ECMWF Coupling sequence and exchange of physical quantities between the atmospheric, ocean wave and ocean models.

Meteorology is also a domain of large data sets. With over 250 PiB of data ECMWF operates the largest archive of meteorological and climate data in the world. This is a very useful resource for scientist and application developers around the world, but often not used because the sheer size of data is hard to manage. The processing was often only limited to users with a large infrastructure. ECMWF is looking for ways to allow users to bring the processing on the large data sets to the data and allow users without large resource to work with it. A first step was the opening of the Climate Data Store in the summer of 2018 to allow users through a web interface and a toolbox to process PiBs of climate data without requiring them to download the data.



This is a new service -- your feedback will help us to improve it

Please note that Tuesday 30th of October, from 9a.m to 10a.m Uk time, a CDS system session will take place. Access and operation to the system might suffer disruptions during this period. Thank you for your understanding.

```
import cdstoolbox as ct


@ct.application(title='Calculate climatologies')
@ct.input.dropdown('variable', values=
['2m_temperature','10m_u_component_of_wind','10m_v_component_of_wind','mean_sea_level
_pressure'])
@ct.input.dropdown('frequency', default='month', values=['dayofyear', 'weekofyear',
'month'], link=True)
@ct.input.dropdown('start', when='dayofyear', default=270, values=range(1, 367))
@ct.input.dropdown('start', when='weekofyear', default=46, values=range(1, 54))
@ct.input.dropdown('start', when='month', default=10, values=range(1, 13))
@ct.output.livefigure()
def compute_climatology(
    variable,
    frequency,
    start
):
    """
    Application main steps:

    - retrieve a variable over a defined time range
    - select a location
    - compute the monthly/daily/weekly climatology and standard deviation
    - show the result as a timeseries on an interactive chart

    """

    data = ct.catalogue.retrieve(
        'reanalysis-era5-single-levels',
        {
            'variable': variable,
```

Figure 9 - Snapshot of the Copernicus Climate Data Store (CDS) toolbox, which allows data processing on the server side.

ECMWF is seeking to find greater use of its model outputs and data sets for its core forecast and data services, as well as the Copernicus services it operates on the behalf of the European

Commission. HiDALGO will look at how users can interact with the services at ECMWF and explore new ways for users to access the output faster and in ways easiest and safest to integrate with their own services. This will build on the latest developments at ECMWF, such as the CDS and the various cloud project it currently builds.

The migration case study (described section 3) and the urban air pollution case study (section 4) will make use of metereological and climate data. Starting with file based delivery of data more advanced ways to couple to these case studies will be explored.

5.2 Telecommunications Simulations

Will couple with: Migration and Social Media Pilots

The telecommunication data for the HiDALGO project is provided by MoonStar Communications (MOON). MOON is a global network service provider that offers voice and messaging termination services and cooperates with leading telecommunication network providers in various regions of the world. MOON is very strong in carrying a high volume of voice traffic originated from and/or terminated to Middle East and African countries. It has direct interconnections with the Tier 1 Network Operators, Mobile Network Operators (MNO), Virtual Mobile Network Operators (MVNO) and various carrier around the world.

The raw data that can be used in the HiDALGO pilots are international Call Data Records (CDR). A CDR is a data record produced by telecommunications equipment that documents the details of a telephone call or other telecommunications transaction (e.g. short text messages SMS) that passes through that facility or device. The record contains various information of a phone call, such as the timestamp when the call is initiated, ringing duration, call duration, completion status (SIP code), source number (A-number), and destination number (B-number).

MOON is operating a network of IP-telecommunication (soft-)switches and session board controllers for the handling of the voice traffic in various countries (see figure 6.4). The switches are connected to each other via secure VPN tunnels. The central controlled dynamic routing of the traffic is based on Quality of Service (QoS) parameters and Least Cost Routing (LCR) tables. This guarantees a balance between an experienced high voice quality with low latency and an optimal cost per minute.

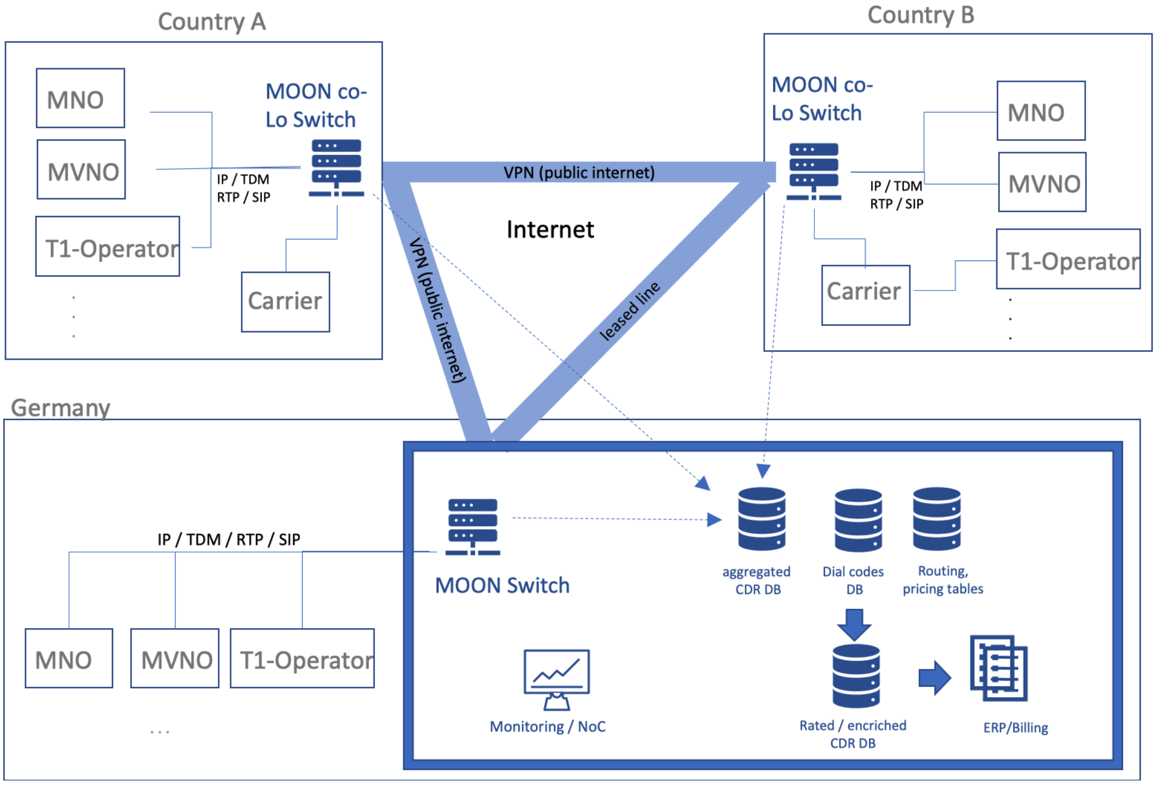


Figure 10 - Architectural overview of the network infrastructure of MOON and where the CDRs are produced and aggregated.

The caller and callee numbers in a CDR are sensitive personal data. This is the reason why the CDRs can not be provided in a raw format to the HiDALGO partners for further processing and analysis. The data first has to be run through an anonymization step.

In addition to the information in the raw data, CDRs can be enriched with:

- geo-location data based on the dial codes,
- origin and destination network operators,
- quality metrics like:
  - Post Dial Delay (PDD): time from the sending of the final dialled digit to the point at which they hear ring tone.
  - Average Call Duration (ACD)
  - Answer-Seizure Ratio (ASR)

An HPDA of the telecommunication data could be performed to detect if calls without an established end-to-end connection and a very short ringing time (aka ping calls) are organic or synthetic. A coupling of the ping calls, UNHCR refugee registration data and voice traffic data with roaming characteristics could give us hints about refugee movements, and we will study this relation in HiDALGO.

The outcome of the HPDA of the telecommunication data could be supporting and/or validating the findings from the results of the migration and social network simulations.

Possible exploitation of these HPDA for MOON could be the forecast of high traffic along refugee routes and transit countries. This would allow MOON to prepare properly for a scale up of hard- and software infrastructure in the corresponding regions. Another valuable exploitation could be the detection of fraud traffic (high volume of synthetic traffic) and to protect against DDoS attacks on the VoIP network.

5.3 Sensor Data Networks

Will couple with: Urban Pollution Pilot

MK, the Hungarian Public Road Non-Profit PLC is responsible for the operation and maintenance of more than 31,000 km national public roads. Road operators are organised in 19 countries at 93 engineering sites managed by the headquarters at Budapest. Activities of the Hungarian Public Road Non-Profit PLC consist of operation, as well as the routine and preventative maintenance of the national public road network including expressways and motorways.

MK has implemented a DATEX Hub in a previous EU-funded project. We are collecting and storing the data in DATEX II format. The Company operates The National Road Databank. This includes up-to -date information on the condition of the national public road network. The program developed for this specific purpose and supervised by the department provides up-to-date information on the traffic flow and accidents incurred on the national public road network and on the condition thereof.

MK offers access to its sensor data networks, which provides input data that is pre-processed and subsequently provided to the Urban Pollution use case. Therefore, MK will provide the legal deployment sites and takes care of the operation and maintenance of the system. The sensor data will be offered from the Traffic Light Management System (proprietary, developer JTR) to the simulations, and output results from the simulations in the use case will be used in turn to guide traffic light control, with the aim to improve the overall air quality.

In the pilot, ARH implements the traffic sensor network. ARH is a global leader in developing and distributing recognition software and ANPR cameras as well as new, all-encompassing ITS (Intelligent Transportation System) solutions. ARH entered the market in 1991 and to date, more than 220 countries worldwide are using the image and data processing know-how of ARH. The company's recognition technologies can be found in every continent, in countries ranging from Australia to Brazil, and from Saudi Arabia to Korea.

The traffic sensor network will provide a range of data. These include, for each vehicle the (1) vehicle category, (2) license plate, (3) speed of vehicles and (4) time of passing vehicle. The sensors also provide (4) its own location and (5) origin-destination information at selected intersection(s) of the road network of Győr, Hungary. Lastly, several sensors will be equipped with (6) air quality sensors to supplement existing operational air quality measurements.

The traffic sensor network connects to a central data center server through IP network and with ARH's EVTS software. This central data server consists of a high capacity hardware and the GLOBESSEY® DATA SERVER – GDS software, which makes massive amounts of traffic data easily accessible for SZE, MK and HPC centres for immediate processing.

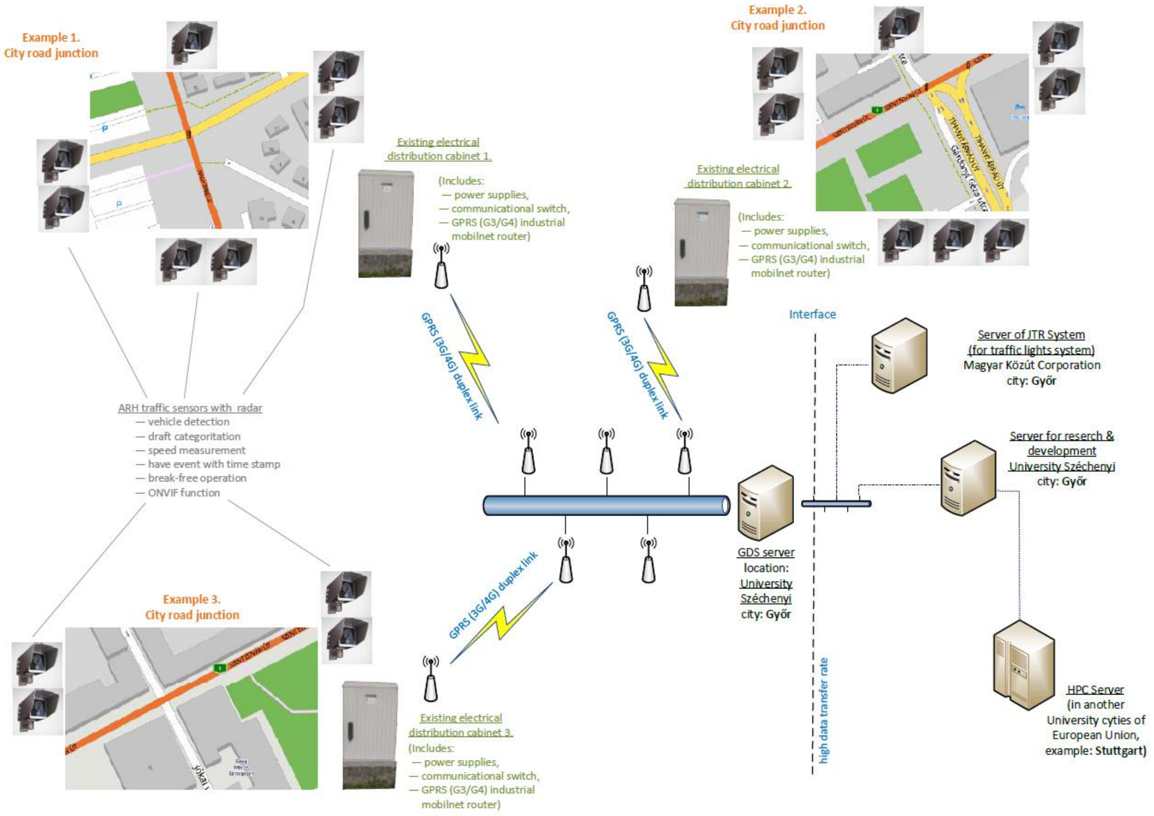

Example 1.
City road junction
Example 2.
City road junction
Existing electrical distribution cabinet 1.
(Includes:
— power supplies,
— communicational switch,
— GPRS (G3/G4) industrial mobilnet router)
Existing electrical distribution cabinet 2.
(Includes:
— power supplies,
— communicational switch,
— GPRS (G3/G4) industrial mobilnet router)
GPRS (3G/4G) duplex link
GPRS (3G/4G) duplex link
Interface
Server of JTR System (for traffic lights system)
Magyar Közút Corporation
city: Győr
ARH traffic sensors with radar
— vehicle detection
— draft categoritation
— speed measurement
— have event with time stamp
— break-free operation
— ONVIF function
Server for reserch & development
University Széchenyi
city: Győr
GDS server
location:
University Széchenyi
city: Győr
Example 3.
City road junction
GPRS (3G/4G) duplex link
high data transfer rate
HPC Server
(in another University cyties of European Union,
example: Stuttgart)
Existing electrical distribution cabinet 3.
(Includes:
— power supplies,
— communicational switch,
— GPRS (G3/G4) industrial mobilnet router)

Figure 11 - Architectural overview of the network infrastructure of ARH.

5.4 Other Models and Data Sources

Besides the above described models/simulations and data sources it is likely that other models and data sources will also be coupled in to the pilot applications. Not all of these are known at this stage of the project, but for instance it is conceivable that food security information from the IPC Global Platform (http://www.ipcinfo.org/) or conflict propagation models could provide a contribution to making refugee movement models more accurate. Moreover, with the increasing popularity of Machine Learning to train models on large existing data sets, some machine-learned models are bound to be tested for integration with one or more of the HiDALGO pilots. HiDALGO wants to embrace these possibilities and will seek active engagement to couple with other models/simulations and data sets/services. To do so clear interfaces need to be defined and ensure that generic and flexible coupling solutions are used [20]. An initial exploration will be done in the first year of the project and presented at M12.

6 Conclusions

In this deliverable we have summarized the initial status of the pilot applications in HiDALGO. This includes the three case studies, but we also have reflected on the coupling models that we seek to integrate within the context of these case studies. All three of the case studies share a strong science case and impact potential, have clearly defined workflows, but differ in priorities in terms of aimed performance and scalability improvements. For the migration pilot, many individual models are available in simplified form, and a major challenge here is to scale up the approach in terms of parallelism, resolution, and range of phenomena incorporated. For the urban air pollution pilot, the main pollution simulation is already in a mature state, but challenges await in incorporating the wide variety of sensor information to make the application more dynamic and suited for optimal use by industry and civic organizations. For the social media pilot, a clear need to analyze the spread of information has been identified, and several key algorithms have been proposed to tackle this problem. Here the development of highly performant and flexible HPDA techniques that can cope with the complex landscape of social media are a key requirement for success. In the case of coupled models, the weather forecasting, telecommunications models, and traffic sensor networks are three priority areas that we will focus on in HiDALGO. That being said, we are also always at the ready to consider the incorporation of additional models, and have already identified first candidates as part of this deliverable.

The contents of this deliverable provide an important reference point for many other activities in HiDALGO, such as the requirements gathering in WP6, the optimization efforts in WP3, and the portal development activities in WP5. However, it is also important to note that important work has already been done to advance the pilots beyond this initial status, and that the contents of this deliverable are expected to be largely superseded once D4.2 has been finalized.

References

[1] UNHCR (2018). *Figures at a glance. United Nations High Commissioner for Refugees.* https://www.unhcr.org/uk/figures-at-a-glance.html, retrieved 2019-03-01.

[2] Salehyan, I. (2014). Forced migration as a cause and consequence of civil war, Routledge, Abingdon.

[3] Schmeidl, S., & Jenkins, J. C. (1998). *The early warning of humanitarian disasters: Problems in building an early warning system*. International Migration Review, 32(2), 471-486.

[4] Suleimenova, D., Bell, D., & Groen, D. (2017). *A generalized simulation development approach for predicting refugee destinations*. Scientific reports, 7(1), 13377.

[5] Groen, D. (2018, June). *Development of a multiscale simulation approach for forced migration*. In International Conference on Computational Science (pp. 869-875). Springer, Cham.

[6] Notman, N. (2017). *City air,* https://www.chemistryworld.com/features/urban-air-pollution/2500224.article, retrieved 2019-02-10.

[7] Stoye, E. (2017). *Government releases plan to tackle nitrogen dioxide pollution,* https://www.chemistryworld.com/news/government-releases-plan-to-tackle-nitrogen-dioxide-pollution/3007256.article, retrieved 2019-02-10.

[8] WHO (2019). *Public health, environmental and social determinants of health (PHE),* https://www.who.int/phe/health_topics/outdoorair/databases/en/, retrieved 2019-02-10.

[9] Reuters (2018). *Most new diesel vehicles exceed emissions limits: German green lobby,* https://www.reuters.com/article/us-germany-emissions-duh/most-new-diesel-vehicles-exceed-emissions-limits-german-green-lobby-idUSKCN1LU1AX, retrieved 2019-02-10.

[10] Gartner (2018). *Confront Key Challenges to Boost Digital Twin Success,* https://www.gartner.com/smarterwithgartner/confront-key-challenges-to-boost-digital-twin-success/, retrieved 2019-02-16.

[11]EU-MATHS-IN (2018). *Modelling, Simulation & Optimization in a Data rich Environment*, https://www.eu-maths-in.eu/EUMATHSIN/wp-content/uploads/2018/05/MSO-vision.pdf, retrieved 2019-02-16.

[12]Blondel et al. (2008); Fast unfolding of communities in large networks, J. Stat. Mech.: Theory and Experiment, 2008(10).

[13]Staudt, Meyerhenke (2016); Engineering parallel algorithms for community detection in massive networks, IEEE Transactions on Parallel and Distributed Systems, 27(1), 171-184.

[14]https://networkit.github.io

[15]Hernandez et al. (2005); SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems, ACM Trans. Math. Software vol. 31(3), 351-362.

[16]Di Napoli et al. (2016); Efficient estimation of eigenvalue counts in an interval, Numerical Linear Algebra with Applications, 23(4), 674-692.

[17]Balay et al. (1997); Efficient Management of Parallelism in Object Oriented Numerical Software Libraries, Modern Software Tools in Scientific Computing, 163-202.

[18]Amestoy et al. (2001); A Fully Asynchronous Multifrontal Solver Using Distributed Dynamic Scheduling, SIAM Journal on Matrix Analysis and Applications, 23(1), 15-41.

[19]Karypis, Kumar (1995); MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 2.0, Available at: http://www.cs.umn.edu/~metis.

[20]Groen, D., Knap, J., Neumann, P., Suleimenova, D., Veen, L., & Leiter, K. (2019). *Mastering the scales: a survey on the benefits of multiscale computing software*. Philosophical Transactions of the Royal Society A, *377*(2142), 20180147.

[21]Varshney, A., Pitchaiah, S. and Armaou, A., 2009. Feedback control of dissipative PDE systems using adaptive model reduction. *AIChE journal*, *55*(4), pp.906-918.