
LOSS MODELLING
FRAMEWORK
Models in Oasis
V1.0
November 2017

OASIS LMF

Models in Oasis

November 2017

40 Bermondsey Street,
London, SE1 3UD
Tel: +44 (0)20 7000 0000
www.oasislmf.org

CONTENTS

Section 1

Introduction

Purpose

The aim of this document is to assist model developers to better understand the principals involved in building and deploying a model in the Oasis Loss Modelling Framework. The document will describe at a high level the various concepts and file formats to describe how a model and exposure data is represented in the Oasis framework.

Background

Oasis is a not-for-profit initiative which provides a community for catastrophe loss modelling based around open architecture, standards, and software. It has the backing of over 40 of the world's leading reinsurers, brokers, and insurers.

Central to the Oasis product set is a "kernel" to calculate ground-up and insured losses to properties using a range of models for hazard and vulnerability. This "kernel" is Open-Source and available to anyone.

Section 2

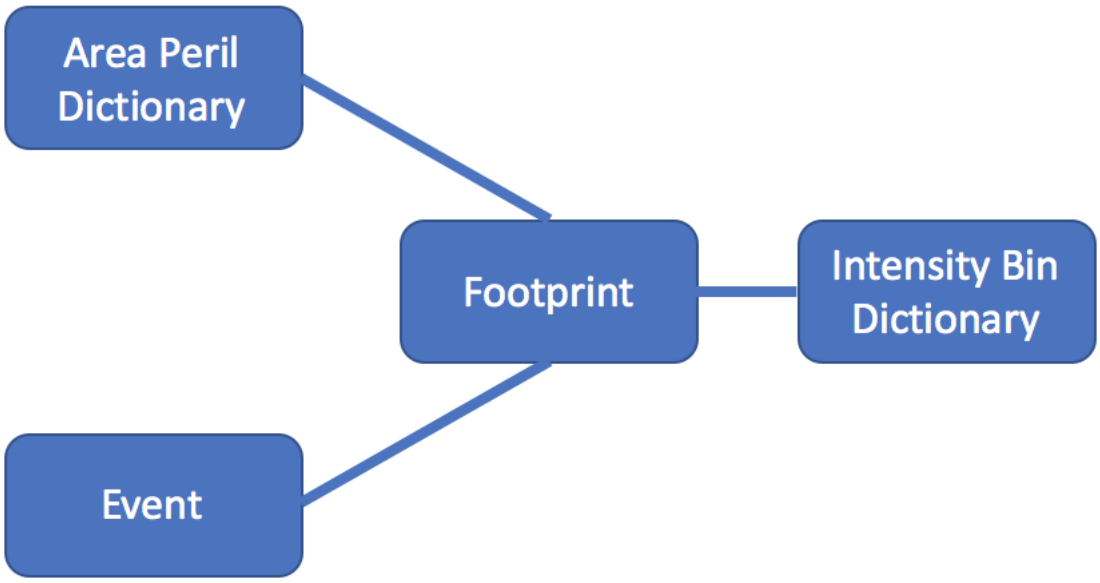
Model Data Format

Overview

A model in Oasis is made up of a hazard module and a vulnerability module. These modules relate to each other via a common intensity metric and relate to exposure data via a set of abstract keys: areaperil_id and vulnerability_id.

Hazard Module

The Hazard Module in Oasis is centered on the **Footprint** file which describes the interaction of **Events** with **Area-Perils**, giving a probability distribution of **Intensity** for each combination.



The main concepts here are:

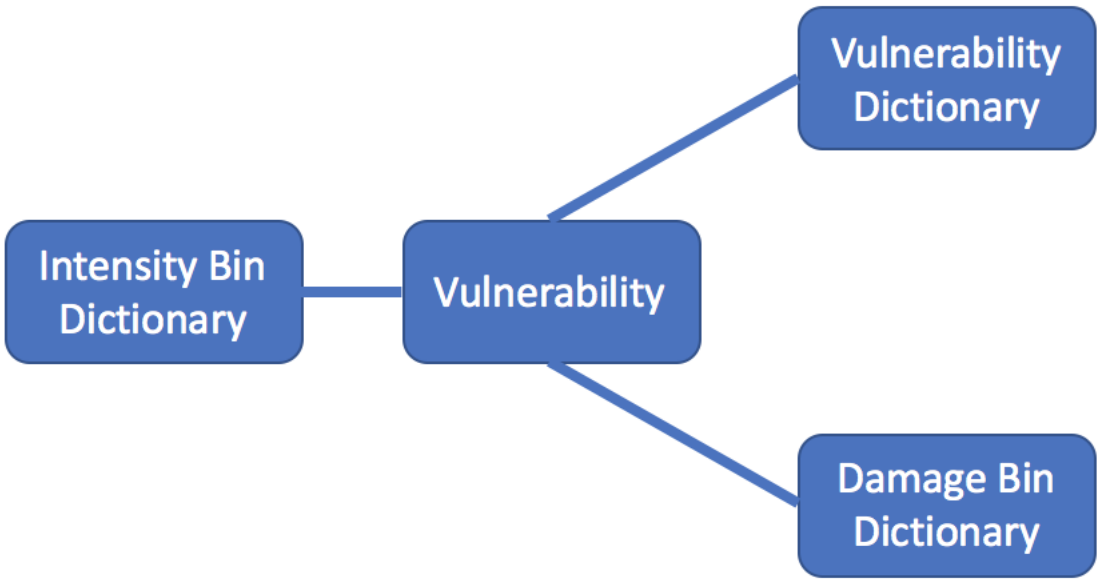
- **Area Peril**: This is an abstract representation of an area for a particular peril. The can be anything – i.e. cells in a grid, polygons, variable resolution grids, point values, administrative regions (e.g. postcodes), etc. The areaperil_id is the key here and needs to be an integer value but what it actually represents doesn't matter for the Oasis calculation.

- **Event**: Again, this is an abstract representation of an actual event (either synthetic or historic) that affects a number of area-perils. This can be a flood, a windstorm, an earthquake, etc. – it doesn't make a difference in Oasis terms. Again, the event_id is the key here and this is simply an integer value that represents that event.

- **Intensity**: The intensity represents a discretised, abstracted set of intensity measures that are specific to the peril (or perils) represented by the events

Vulnerability Module

The Vulnerability Module in Oasis is centered on the **Vulnerability** file which describes the interaction of **Intensities** with **Vulnerability types**, giving a probability distribution of **Damage Ratio** for each combination.

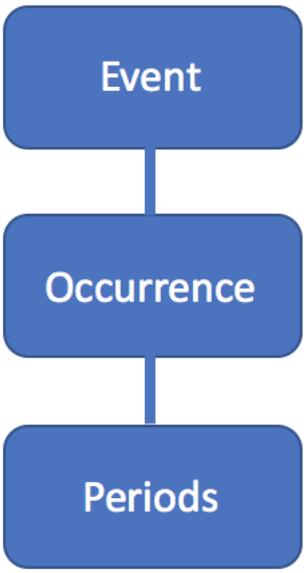


The main concepts here are:

- **Intensity**: The intensity represents a discretised, abstracted set of intensity measures that are specific to the peril (or perils) represented by the events
- **Vulnerability Dictionary**: Again, this is an abstract representation of various vulnerability functions. These can be as sophisticated as required for the model. The vulnerability_id is the key here and this is simply an integer value that represents that vulnerability function.
- **Damage**: The Damage represents a discretised, abstracted set of damage ratios.

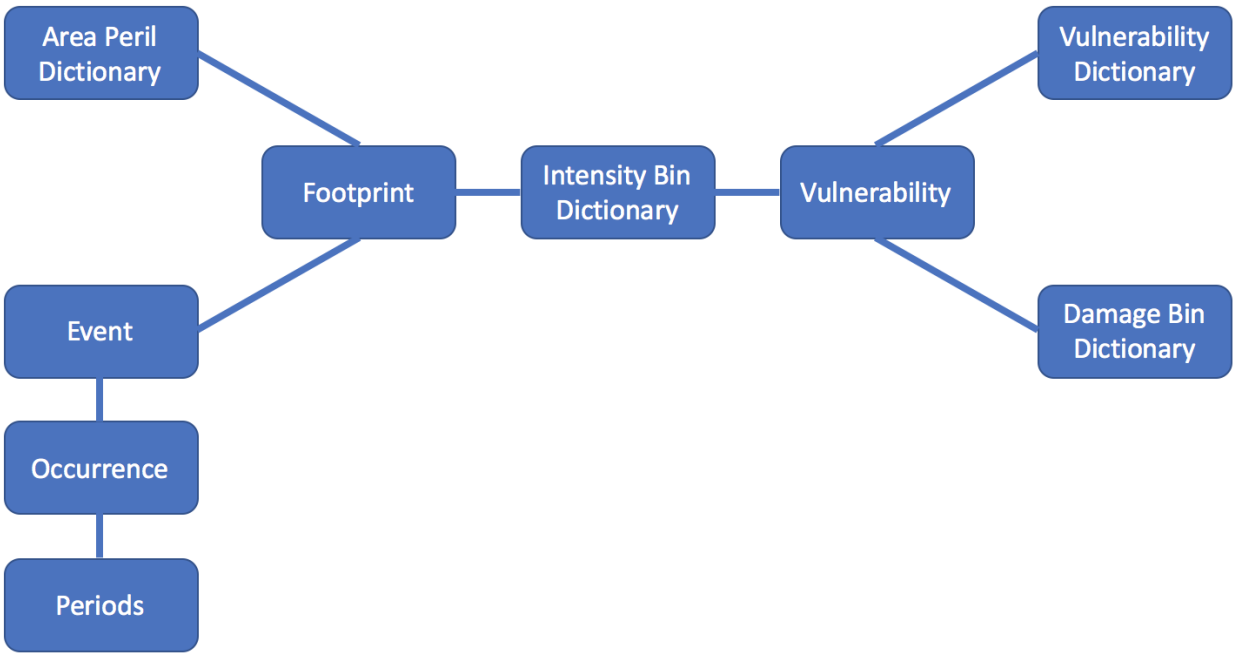
Occurrences

The final part of the model data definition in Oasis is the Occurrence file. This file details the occurrences of events over time and is used in time based outputs such as Average Annual Loss and Loss Exceedance Curves. An optional extension to the Occurrence file is the Periods file which allows a weighting to be placed on occurrences.

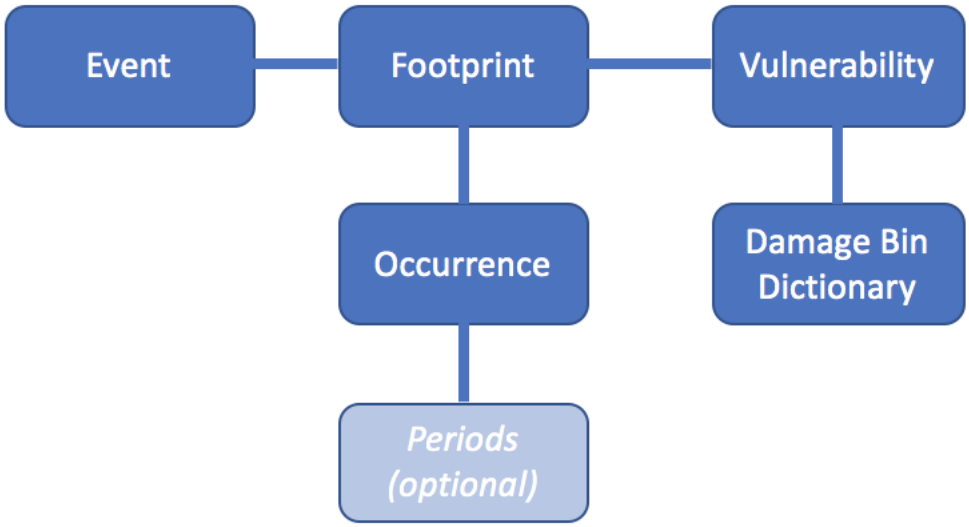


Putting it all together

Now that we have the hazard module, the vulnerability module and the occurrence definitions, we can put them all together across a common intensity bin definition to create a complete representation of a model in Oasis.



It should be noted that some of these files are not required for the oasis calculation and are simply in place to provide a reference to what the abstracted ids mean in reality. The complete picture with required files looks like this:



For examples of simple model implementations, see Appendix A.

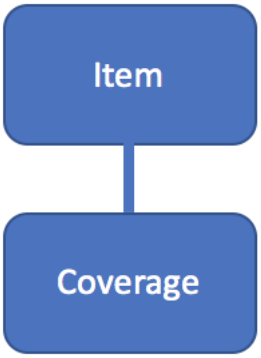
Section 3

Exposure Data

Exposures

Exposure data is also represented in an abstracted form in Oasis. The individual exposed coverages are depicted as integer “items” where an item is a location, peril, coverage type combination. Item is the lowest level of exposure representation in Oasis terminology. Each item will have an areaperil_id to represent its position in the geography of the model and a vulnerability_id to identify the type of exposure that it is. The item file also includes a “group_id” field which is used for correlation across items in the oasis calculation. Items with the same group_id will be subject to the same random number when sampling the effective damage distribution during the simulation calculation.

One or many items can form a “coverage” which is a location, coverage-type combination in Oasis terms. The total insured value (TIV) is stored at this level in the data representation of exposure against the “coverage_id”.

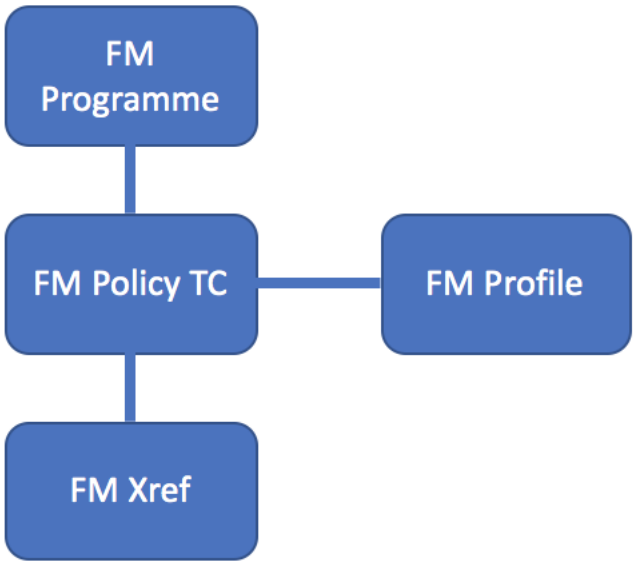


Financial Module

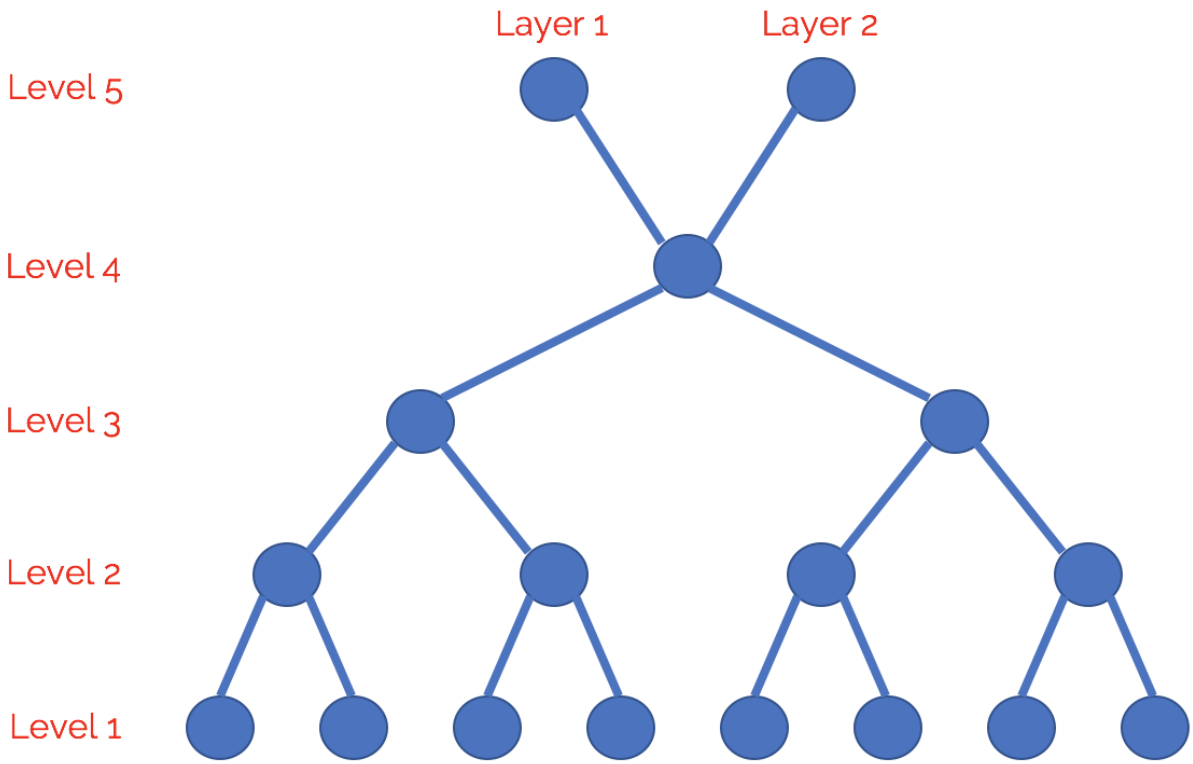
Financial terms and conditions are also represented in an abstracted manner in Oasis using three main files:

- FM Programme which describes the hierarchical aggregation of financial structures
- FM Profile which describes the calculation rules and the values to be used
- FM Policy TC which connects the hierarchical aggregation structure to the profile and also describes any insurance layer conditions

The final file in the Financial Module is the FM Xref which is used to connect the 3 core financial module files to the output summary files (see later).



The FM structure is required to be hierarchical through levels until the final level, at which point a layer structure can be implemented, so that a single insurance contract can be applied over many layer applications. It is not possible to have lower levels split out to higher levels, they must aggregate through the levels.



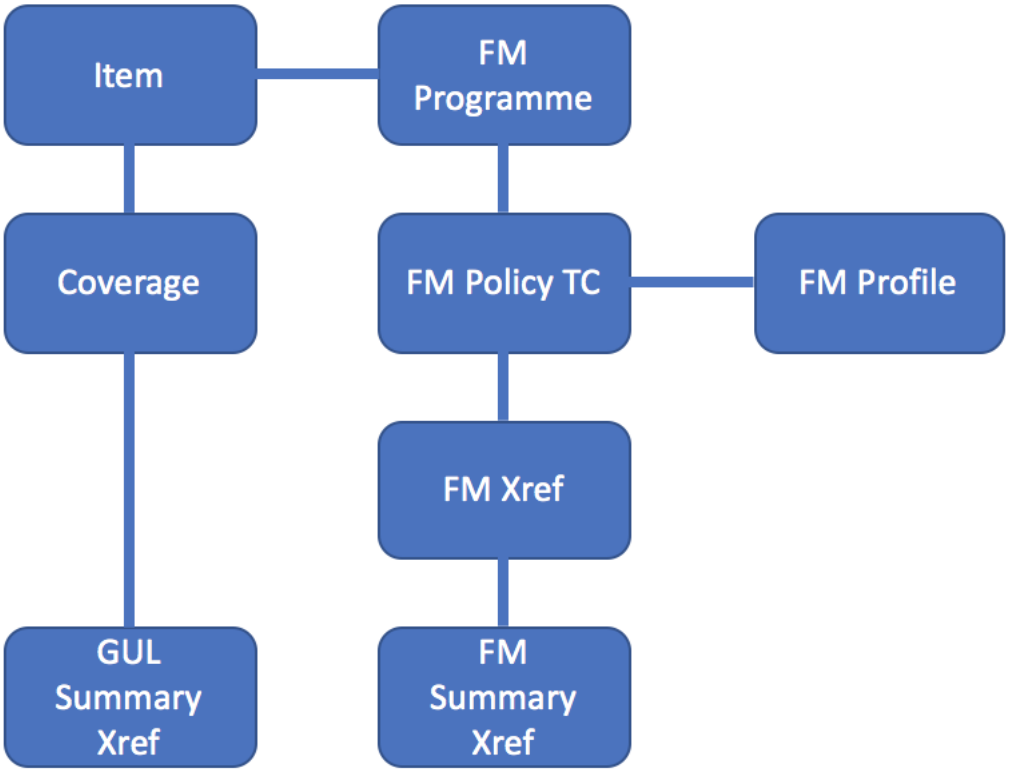
Summary Level Files

Finally, there are two files that define how any outputs are aggregated:

- GUL Summary Xref defines how coverage ids should be aggregated in any Ground Up Loss outputs.
- FM Summary Xref defines how FM Output IDs are aggregated in any Insured Loss outputs.

Putting it all together

The diagram below shows how the different exposure data files should fit together:

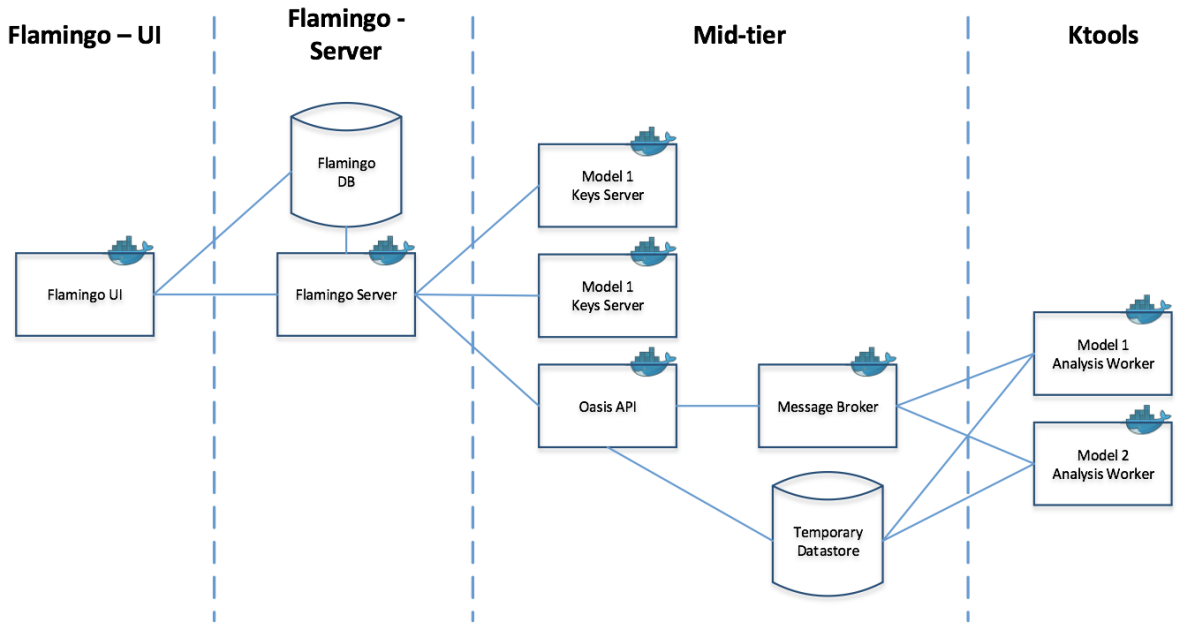


Section 4

Architecture

The standard Oasis architecture is made up of four tiers:

- Flamingo UI – this is the user interface and is written in R Shiny. The Flamingo UI provides an example web front end to Oasis that can be used to load exposure data, transform that exposure data into the Oasis formats described above and request various analyses against models with this data.
- Flamingo Server – the Flamingo server manages the requests that are passed form the Flamingo UI, converts data and requests form the Flamingo UI to the Oasis back end and stores information about the requests and the data in the Flamingo DB
- Mid-tier – the Oasis Mid-tier acts as a message broker and queue management system for jobs into the Oasis back end where the calculations are executed
- Ktools – this is the calculation kernel where the simulations are executed. Kit is possible to have many calculation workers hook into a single message broker.

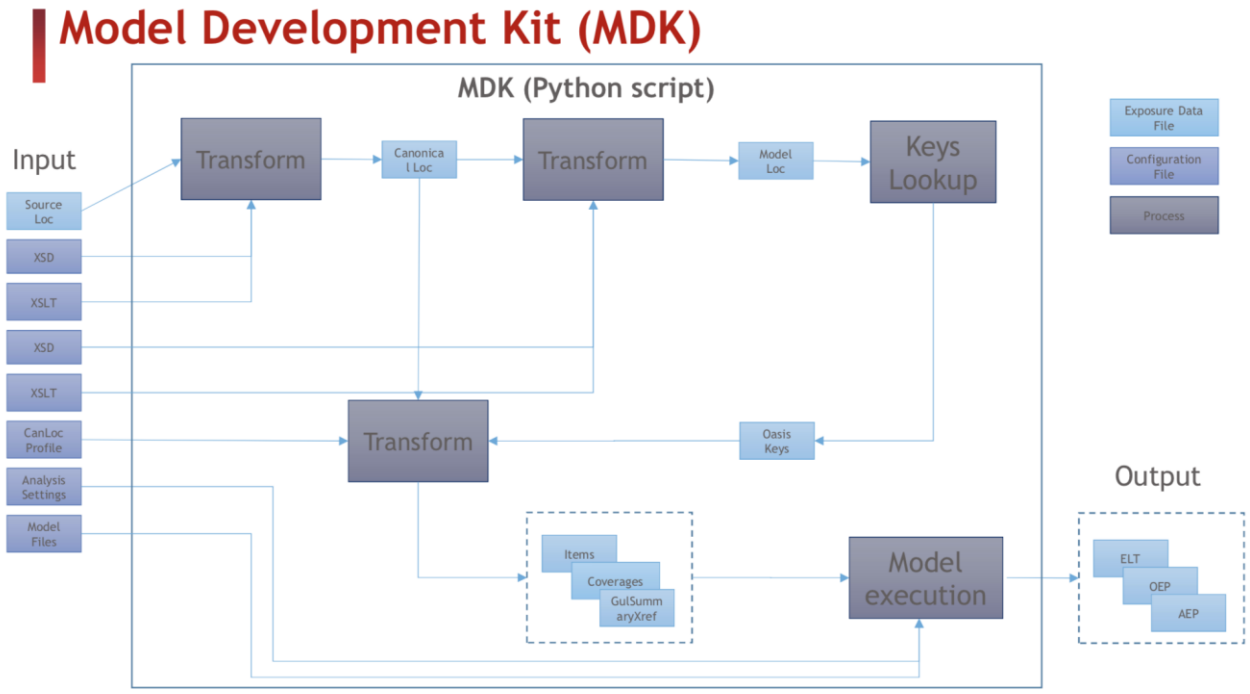


Component	Description	Technology
Flamingo Shiny	Browser based application for managing exposure data and operating modelling workflows	R Shiny, ShinyProxy
Flamingo Server	Services for interacting with exposure and output data	Flask
Flamingo Database	Storage for exposure data, workflow configurations and system data.	SQL Server
Oasis API	Services for uploading Oasis files, running analyses and retrieving outputs.	Flask, Celery
Message Queue	Queues for managing workload across multiple calculation back ends.	Rabbit MQ
Data store	Storage for transient analysis data.	File share
Keys Server	Model specific services for generating area peril and vulnerability keys for a particular portfolio.	Flask
Calculation Backend	Executes a model.	Celery, running as daemon, ktools

Section 6

Model Development Kit

The Model Development Kit is intended to replicate all of the elements of a full deployment of Oasis but on a single development machine. It does not provide a user interface but is accessible from the command line and it allows model developers to build and test all of the elements of a full oasis deployment without needing a fully deployed system.



It provides a suite of software tools that make it easier and more cost effective to migrate models into the Oasis framework. Phase 1 (Jan 2018) will provide software that allows a model developer to easily test models using existing test exposure sets. Command line tools (Python) can run models directly from standard, insurance exposure files

Appendix 1

Example Models

Example 1: UK Windstorm Model

Example 1 shows a small, example windstorm model with two events affecting two area perils. The areas are defined as point lat/long positions and there are three intensity bins representing wind-speeds 50-60, 60-70 and 70-80 m/s. There are two vulnerability functions defined for the model based on construction schemes. All of the secondary uncertainty in the model is represented in the vulnerability functions.

areaperil_dict

areaperil_id	lat	lon
1	51.5	0.13
2	53.81	3.03

footprint

event_id	areaperil_id	intensity_bin_index	prob
1	1	1	1
1	2	1	1
2	1	2	1
2	2	3	1

intensity_bin_dict

bin_index	bin_from	bin_to	interpolation	interval_type
1	50	60	55	1202
2	60	70	65	1202
3	70	80	75	1202

event

event_id
1
2

occurrence

event_id	period_no	occ_year	occ_month	occ_day
1	1	1	9	3
2	1	1	10	18
1	4	4	8	16
2	10	10	9	24

vulnerability_dict

vulnerability_id	construct_scheme	construct_code
1	ATC	1
2	ATC	2

vulnerability

vulnerability_id	intensity_bin_index	damage_bin_index	prob
1	1	1	0.9
1	1	2	0.1
1	2	1	0.8
1	2	2	0.2
1	3	1	0.7
1	3	2	0.3
2	1	1	0.8
2	1	2	0.2
2	2	1	0.5
2	2	2	0.5
2	3	1	0.1
2	3	2	0.9

damage_bin_dict

bin_index	bin_from	bin_to	interpolation	interval_type
1	0	0.4	0.2	1202
2	0.4	1	0.7	1202

An example loss calculation for an exposure in London with building code ATC1 and a Total Insured Value of £100,000 might go:

- The exposure is in London and so is classed as being in areaperil_id = 1 and has building code ATC1 so has vulnerability function = 1.
- Being in areaperil 1 means the exposure is subject to the following events:
    - Event 1, intensity bin 1 (50-60m/s), with probability 1 and occurrences in period 1 and 4
    - Event 2, intensity bin 2 (60-70m/s) with probability 1 and occurrences in period 1 and 10
- Two random samples are simulated against the model:
    - For Event 1:
        - Sample 1 has random number 0.1265 which gives a damage bin index of 1 (since 0.1265 is between 0 and 0.9) which leads to a 0.2 damage factor or £20,000 loss

        - Sample 2 has random number 0.5643 which also gives a damage bin index of 1 (since 0.5643 is also between 0 and 0.9) which leads to a 0.2 damage factor or £20,000 loss

    - For Event 2:

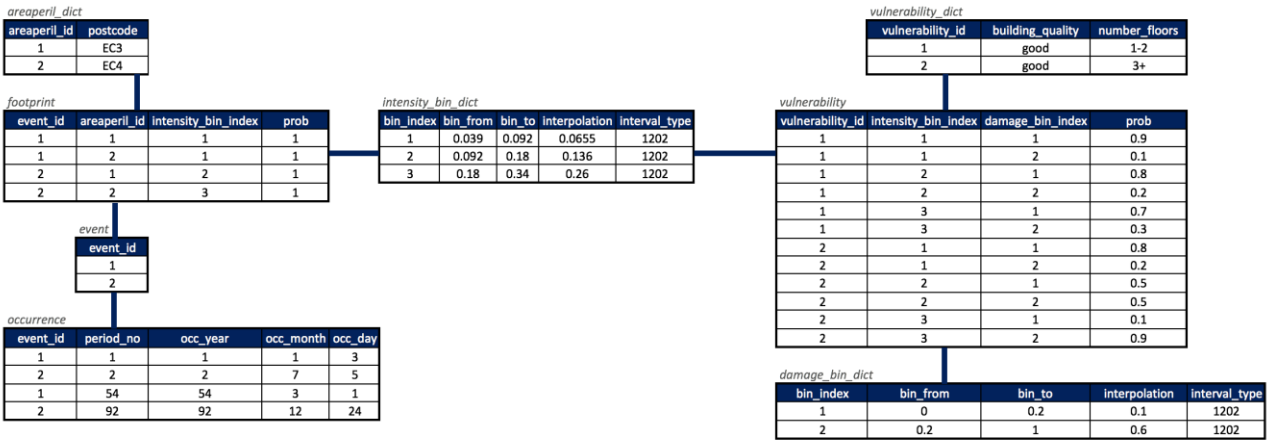
        - Sample 1 has random number 0.3444 which gives a damage bin index of 1 (since 0.3444 is between 0 and 0.8) which leads to a 0.2 damage factor or £20,000 loss

        - Sample 2 has random number 0.8101 which gives a damage bin index of 2 (since 0.8101is between 0.8 and 1.0) which leads to a 0.7 damage factor or £70,000 loss

- So, we end up with four sample outputs and an ELT might look like: Event 1 has sample mean loss £20,000; Event 2 has sample mean loss £45,000

Example 2: UK Earthquake Model

Example 2 shows a small, example earthquake model with two events affecting two area perils. The areas are defined as postcodes and there are three intensity bins representing peak ground accelerations 0.039-0.092, 0.092-0.18 and 0.18-0.34 g. There are two vulnerability functions defined for the model based on building quality and number of floors. All of the secondary uncertainty in the model is represented in the vulnerability functions.



areaperil_dict

areaperil_id	postcode
1	EC3
2	EC4

footprint

event_id	areaperil_id	intensity_bin_index	prob
1	1	1	1
1	2	1	1
2	1	2	1
2	2	3	1

intensity_bin_dict

bin_index	bin_from	bin_to	interpolation	interval_type
1	0.039	0.092	0.0655	1202
2	0.092	0.18	0.136	1202
3	0.18	0.34	0.26	1202

vulnerability_dict

vulnerability_id	building_quality	number_floors
1	good	1-2
2	good	3+

vulnerability

vulnerability_id	intensity_bin_index	damage_bin_index	prob
1	1	1	0.9
1	1	2	0.1
1	2	1	0.8
1	2	2	0.2
1	3	1	0.7
1	3	2	0.3
2	1	1	0.8
2	1	2	0.2
2	2	1	0.5
2	2	2	0.5
2	3	1	0.1
2	3	2	0.9

damage_bin_dict

bin_index	bin_from	bin_to	interpolation	interval_type
1	0	0.2	0.1	1202
2	0.2	1	0.6	1202

event

event_id
1
2

occurrence

event_id	period_no	occ_year	occ_month	occ_day
1	1	1	1	3
2	2	2	7	5
1	54	54	3	1
2	92	92	12	24

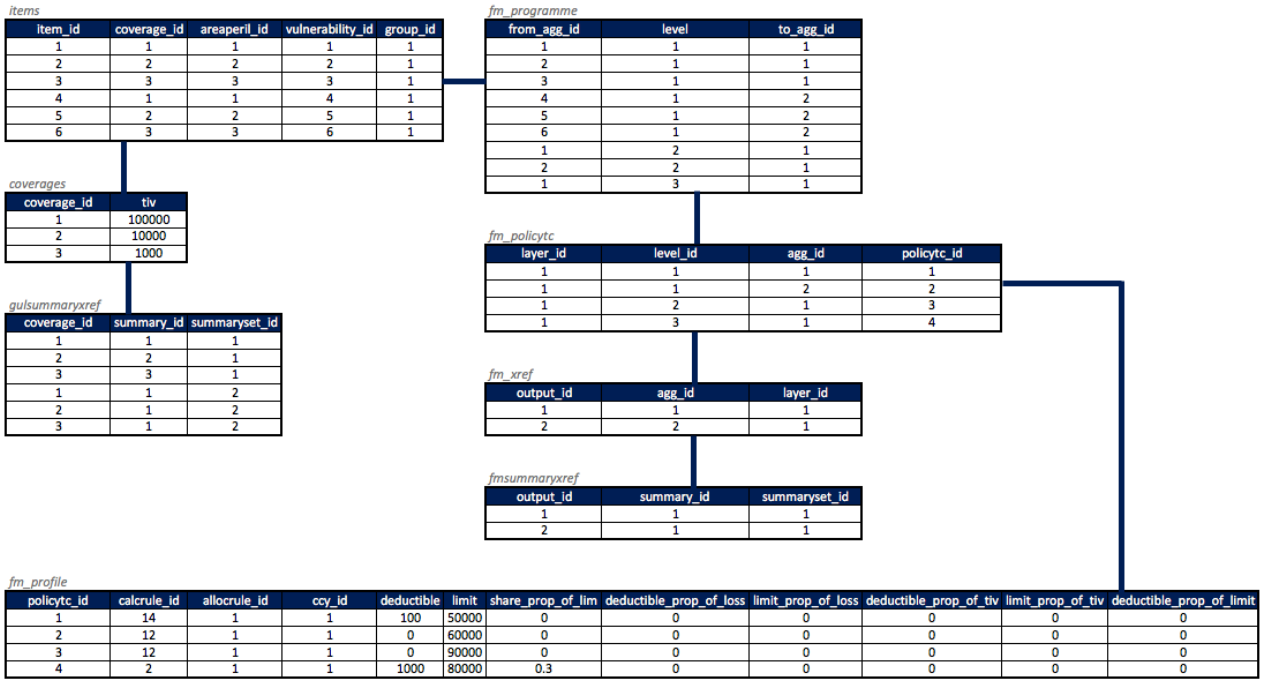
You will notice that the structure for these two models is the same and, in fact, the model data itself is exactly the same as for the windstorm model. In example 1. The only things that have changed are the dictionary definitions of the abstract ids (to represent different measures) and the occurrence file values (to represent a longer period).

Appendix 2

Example Exposures

The example below shows an example exposure structure in the Oasis format. Key points to note are:

- There are six items which represent (in this case) three coverage types across two sub-perils for a single location
- The total insured value (tiv) is stored at coverage level and any losses for the coverage will be capped at this point
- The output aggregation levels are defined by gulsuymmaryxref – which in this case are coverage level (for summary set 1) and overall (for summary set 2) and fmsummaryxref for policy level outputs (here set to overall only)
- The FM Programme table shows how the item losses will be aggregated up over three financial levels
- The FM Policy TC table connects to the FM Profile table to show the financial terms that will be applied to these aggregation levels in order, including the calculation rules to be applied.



items

item_id	coverage_id	areaperil_id	vulnerability_id	group_id
1	1	1	1	1
2	2	2	2	1
3	3	3	3	1
4	1	1	4	1
5	2	2	5	1
6	3	3	6	1

coverages

coverage_id	tiv
1	100000
2	10000
3	1000

gulsummaryxref

coverage_id	summary_id	summaryset_id
1	1	1
2	2	1
3	3	1
1	1	2
2	1	2
3	1	2

fm_programme

from_agg_id	level	to_agg_id
1	1	1
2	1	1
3	1	1
4	1	2
5	1	2
6	1	2
1	2	1
2	2	1
1	3	1

fm_policytc

layer_id	level_id	agg_id	policytc_id
1	1	1	1
1	1	2	2
1	2	1	3
1	3	1	4

fm_xref

output_id	agg_id	layer_id
1	1	1
2	2	1

fmsummaryxref

output_id	summary_id	summaryset_id
1	1	1
2	1	1

fm_profile

policytc_id	calcrule_id	allocrule_id	ccy_id	deductible	limit	share_prop_of_lim	deductible_prop_of_loss	limit_prop_of_loss	deductible_prop_of_tiv	limit_prop_of_tiv	deductible_prop_of_limit
1	14	1	1	100	50000	0	0	0	0	0	0
2	12	1	1	0	60000	0	0	0	0	0	0
3	12	1	1	0	90000	0	0	0	0	0	0
4	2	1	1	1000	80000	0.3	0	0	0	0	0

Appendix 3

GitHub Repositories

All of the Oasis LMF code is available on the Oasis LMF GitHub page https://github.com/OasisLMF but the following areas of the repositories are particularly useful to better understand and expand on the concepts discussed in this document:

- https://github.com/OasisLMF/ktools is the ktools repository itself where all of the ktools code is stored.
- Of interest to understand the concepts is the docs section https://github.com/OasisLMF/ktools/tree/master/docs/md and specifically:
    - https://github.com/OasisLMF/ktools/blob/master/docs/md/DataConversionComponents.md which describes the data formats and the tools to convert data in and out of the binary formats that are required for the calculation kernel
    - https://github.com/OasisLMF/ktools/blob/master/docs/md/CoreComponents.md which describes the core components that are required to run a model once the data is in the correct format
    - https://github.com/OasisLMF/ktools/blob/master/docs/md/FinancialModule.md describes the financial module including the calculation rules that are available.
    - https://github.com/OasisLMF/ktools/blob/master/docs/md/Workflows.md describes the workflow elements that are involved in running the Oasis calculations and shows some example workflows.
- https://github.com/OasisLMF/omdk is the Model Development Kit repository