

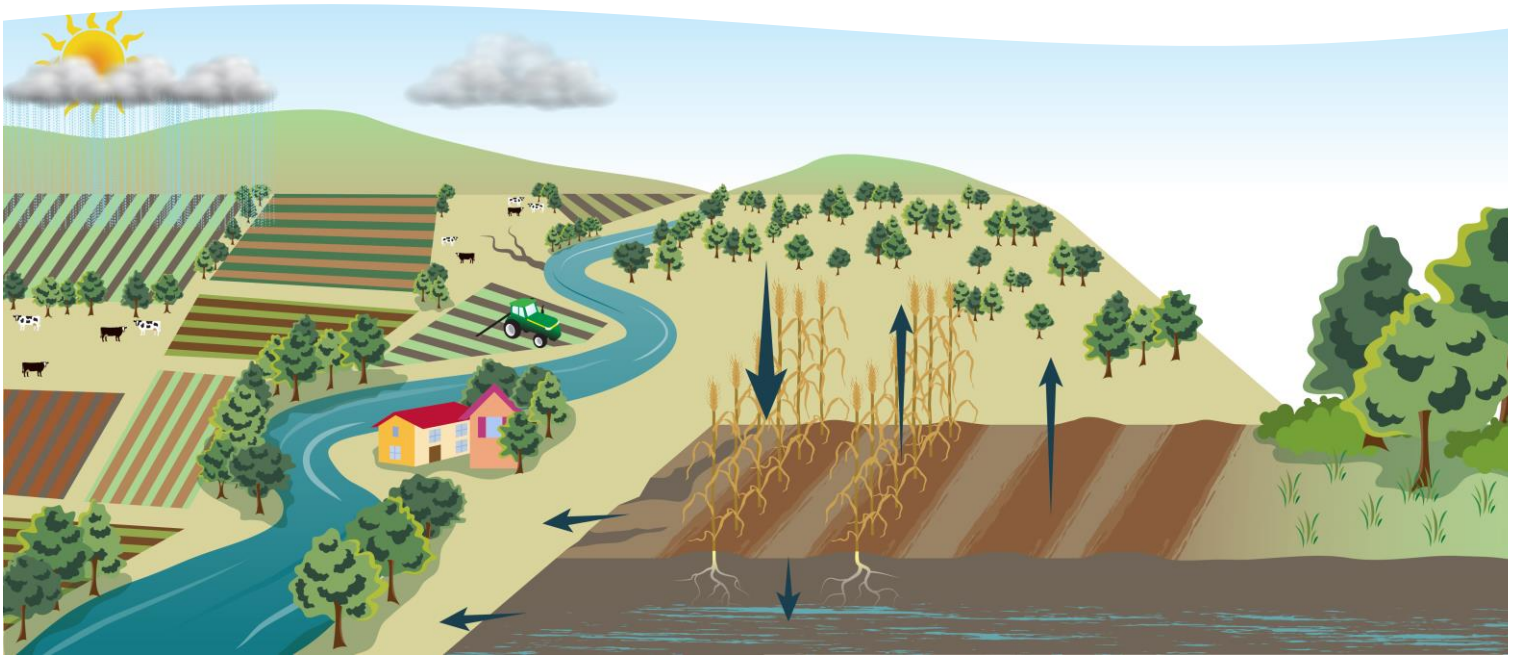
Developing HowLeaky Platform for Improved Governance, Facilitation of Development, and Open Reproducible Science

August 2019 – Final report

Afshin Ghahramani

Roy Anderson

Al Doherty



USQ CSAS Publication 1006688-19
Centre for Sustainable Agricultural Systems
Institute for Life Sciences and Environment
University of Southern Queensland
Toowoomba, Queensland, Australia, 4350

Copyright and disclaimer

Printed in October 2019 by the USQ Centre for Sustainable Agricultural Systems, Toowoomba. Material from this publication is subject to copyright and may not be used, in whole or part, unless formal acknowledgement to the research institution is made. This report is the work of the author and does not represent the views or policies of the Queensland Government.

Citation

Ghahramani, A., Anderson, R., Doherty, A. (2019) Developing HowLeaky Platform for Improved Governance, Facilitation of Development, and Open Reproducible Science. USQ Centre for Sustainable Agricultural Systems. Final technical report for the Department of Environment and Science, Queensland Government.

Acknowledgement

This project was supported by funding from the Queensland Government Department of Environment and Science (DES) under a partnership between DES and University of Southern Queensland. Support was provided by HowLeaky stakeholders since the start of the project these include David Freebairn, Paul Lawrence, David McClymont, Keith Pembleton, Dan Rattray, Jenny Riches, Brett Robinson, Mark Silburn, Sarah Stevens, and Craig Thornton.

The logo, figure in the front page of the report, and HowLeaky website is designed by [hodgeenvironmental](#).

Enquiries should be addressed to:

Afshin.Ghahramani@usq.edu.au

Institute for Life Sciences and the Environment | Centre for Sustainable Agricultural Systems

University of Southern Queensland

Toowoomba, Queensland, 4350

Contents

Executive summary	5
1 Background.....	6
2 New HowLeaky and impact on the end user.....	8
3 Re-development	10
4 Application programming interfaces (APIs).....	16
5 Code development and auto-validation	19
6 HowLeaky Panel.....	21
7 Code change submission and automated review system.....	23
8 Dataset management	24
9 Access to the model parameters	28
10 IP & Licence	30
11 Terms of Reference	31
12 Future development opportunities	31
Appendices	35
Appendix 1 Hosting infrastructure.....	35
Appendix 2 GitHub Forking Workflow	38
Appendix 3: Jenkins Overview.....	41

Figures

Figure 1 Structure of HowLeaky showing main model, sub-models, inputs and outputs (Queensland Government, 2019).....	12
Figure 2 Structure of daily simulation flow in HowLeaky.	13
Figure 3 The public HowLeaky code repository in GitHub.	14
Figure 4 The Jenkins page of the HowLeaky project.	15
Figure 5 Identifying 'Authenticate' Swagger endpoint.	16
Figure 6 Operation and results from 'Authenticate' Swagger endpoint.	17
Figure 7 Location of the 'Authorise' button in the HowLeaky Swagger interface.	18
Figure 8 Sample input for Authorisation on the HowLeaky Swagger interface. ..	18
Figure 9 Architecture of the HowLeaky dev-ops system for model development.	19
Figure 10 The process of development and code submission.	20
Figure 11 Interaction between HowLeaky stakeholders (at the time of initiating original Panel).....	22
Figure 12 The processes of a code submission and automated review for HowLeaky.	23
Figure 13 Distribution of datasets across Australia.....	24
Figure 14 Entity Relationship Diagram of the HowLeaky database.....	28

Figure 15 The model parameters of the models datasets are accessible via simulation menu. The parameters of the core validation site are accessible and changes are subject to approval frn the Panel. 30

Tables

Table 1 Description of developed items during the project and how to access them. 6

Table 2 A comparison of the new HowLeaky with the old version and also with APSIM as an alternative model..... 10

Table 3 Description of core datasets for verification HowLeaky code’s re-development. AAR: average annual rainfall. PAWC: Plant available water capacity. Datasets are accessible at www.howleaky.com. 26

Executive summary

HowLeaky is an important asset for the Queensland Government, universities, research institutes and is useful for estimating water balance and water quality for a range of land uses and management options. It is applied to benchmarking water quality changes associated with the Great Barrier Reef catchment investments.

This project has developed a new web-based platform for the HowLeaky model to facilitate improved management, development and governance of the model. This was delivered through a collaborative research agreement between the University of Southern Queensland and the Queensland Government, Department of Science and Environment.

With these enhancements and improved accessibility and management, HowLeaky becomes a community governed model, with software, source code, and experimental datasets required for development being accessible and freely available through the HowLeaky website (www.howleaky.com). Tasks completed during this project included:

1. Model code translated from C++ to C# and a new command line version of HowLeaky developed;
2. A hosting framework developed with model source code placed on GitHub while the web-based platform and databases are on three servers hosted by Nectar (The National eResearch Collaboration Tools and Resources project) infrastructure;
3. Code and model parameter change submission process was automated;
4. A web-based interface developed to facilitate code modifications and automate model validation using key datasets and parameters of calibrated models;
5. Available datasets and parameters archived on the website and available to users;
6. Workflow developed to facilitate access and archival of data and parameters within the hosting framework;
7. Terms of reference and peer review procedures developed, documented and agreed;
8. An oversight committee convened for peer review;
9. The model can be run via the website, building on previous examples and having access to external databases of climate (P51 from SILO) and soil descriptions (Soil and Landscape Grid of Australia) using Application Programming Interfaces (APIs); and
10. The model has a Swagger UI for HowLeaky REST API which can be accessed by third parties for review, using related API the simulations can be run on the cloud.

Given the nature of developing new software, this development should be regarded as a first phase of an ongoing development process, while completing all deliverables of the project. The software has not yet been tested with a user group to determine the user reaction or system robustness (was not part of the project). It is expected that future iterations of software development will be required following feedback received from both modellers and developers. Additional time, effort and resources may be required to ensure model uptake of this version of HowLeaky within the Queensland Water Modelling Network.

A peer review and support framework including a HowLeaky Steering Committee and Panel were developed to oversee model peer review activities and governance. The Panel includes members from the Queensland Government, the University of Southern Queensland and the private sector. The activities of the Panel are overseen by a Steering Committee which consists of the delegate(s) from the Queensland Government and the University of Southern Queensland. The Steering Committee ensures on-going support for the development and use of HowLeaky.

The project team has explored the required further developments and significant opportunities identified for future model developments and implications. Table 1 presents project deliverables and provides access links to them on the website and related web services.

Table 1 Description of developed items during the project and how to access them.

Milestone items	Deliverable	Access
Code change submission and automated review system functioning	Completed and functioning in the form of codes	placed in GitHub & build server https://GitHub.com/HowLeaky/HowLeaky.git http://howleakybuild.net.au/
Data and parameters able to be archived within the hosting framework	Completed, MySQL database	MySQL DB on HowLeaky Data server Server IP: provided upon request Username: provided upon request Password: provided upon request
Identification, collation and evaluation of key data sets for model validation	Completed, placed in database	In MySQL database in HowLeaky Data server
Model interface for the submission of code changes, auto-validation and archiving of parameter sets	Completed, website	www.Howleaky.com
Data sets and parameters available for access within the hosting framework	Completed, access through new website	www.Howleaky.com
Terms of reference and peer review procedures developed, documented and agreed	Completed, documented and agreed by DES and USQ	http://www.howleaky.com/library/HowLeakyTOR_2019.pdf
Model and parameter sets accessible and interface operational through hosting site	Completed	www.Howleaky.com
HowLeaky model peer review and oversight committee convened	Completed	To be placed on the HowLeaky website.

1 Background

HowLeaky is a one dimensional water balance model with sub-models for soil erosion, phosphorus, pesticide dynamics, and irrigation to simulate the quality of water leaving agricultural systems at the paddock or field scale. The modelling environment is built on

the foundations of the PERFECT model (Littleboy et al., 1992). PERFECT has been strongly influenced by CREAMS (Knisel et al., 1980) and EPIC (Williams, 1983).

The water balance base-model of HowLeaky estimates the volume of water in each soil layer on a daily time step for a single point with a one-dimensional flux movement. It is a bucket model whereby each soil layer is considered as a bucket, and the maximum capacity of each layer is defined by saturated water content (SAT). A downward flux movement occurs when soil water exceeds the drained upper limit (DUL) of the soil layer using a maximum drainage rate parameter.

HowLeaky calculates soil erosion based on daily runoff amount and a sediment concentration-cover relationship (Freebairn and Wockner 1986). The model predicts soil loss by accounting for changes in ground cover and runoff, both factors that can be controlled through management. Universal Soil Loss Equation (USLE) factors for erodibility (K, metric units), slope-length factor (LS), and practice factors from Wischmeier and Smith (1978) are also required for estimation of erosion. HowLeaky quantifies phosphorus (P) export from the paddock which is sourced from native soil minerals; fertilisers and manures are not modelled except in how they change the soil P status. The N sub-model of HowLeaky is still under development and currently it can only simulate the transport of nitrate in three separate transport mechanisms from the system. These are through dissolved nitrate in the runoff, leaching, and particulate transport by runoff. There are also multiple options for each transport method implemented in the sub-model. The pesticides sub-model has evolved from CREAMS/GLEAMS (Leonard et al., 1987). The improvements in the model have been based on the field experiments conducted in Australia (e.g. Silburn, 2003; Shaw et al. 2011). Pesticides can be applied to the plant canopy, crop stubble or the soil, from which they can be washed off by rainfall to the layer below. Degradation in each pool uses an exponential decay function dependent on temperature. Loss of pesticide in runoff is based on an empirical relationship between pesticide concentration in soil and concentration in runoff (Silburn, 2003), while the partitioning of each chemical into the water and soil phases is determined by a linear isotherm. Application of chemicals below the soil surface and leaching are not simulated.

Model capability and application

The HowLeaky model supports a better understanding of the landscape water dynamics and implications of land management practices. The model is a critical tool for both the Department of Natural Resources, Mines, and Energy (DNRME) and the Department of Environment and Science (DES). The model has applications for policy making within the

Queensland Government (Carroll and Yu, 2018). It is being used in the “Paddock to Reef Integrated Monitoring, Modelling and Reporting Program” (Carroll et al., 2012) to evaluate potential improvements in water quality of water bodies in the catchments entering into reef through simulation of best management practices at paddock scale (Shaw and Silburn, 2016). It is the main model within the Paddock to Reef modelling program that provides the required time series of paddock scale modelling into the Source Catchments model which ultimately informs the Reef Report Card (The Queensland Government, 2016). Outputs from the HowLeaky model are also being used to evaluate the cost-effectiveness of paddock scale land management practices (Department of Agriculture and Fisheries' Cost Effectiveness Calculator), and provides updates on the SafeGauge decision support tool for nutrient management in sugar cane. In addition to the modelling power, HowLeaky has an advantage in computational speed with the capability of running long term simulations almost instantly. This is due to the simplicity of the computations and the computational efficiencies implemented in the code. This capability provides opportunities to develop tools for smartphones and other mobile devices to use HowLeaky as their modelling engine while maintaining the model's performance. For example, SoilWaterApp (Freebairn et al., 2018) developed for virtual monitoring of soil moisture in cropping systems, it is smart phone app and is well adopted by farmers across Australia.

The HowLeaky model can be efficiently used to run large factorial simulations in batch mode, enabling an evaluation of a matrix of soil, climate and management practices. As a result, HowLeaky has a comparative advantage over other soil water balance models when conducting modelling projects that require significant computing resources.

2 New HowLeaky and impact on the end user

The HowLeaky model is being used by the Queensland Government, universities, industry, and other parts of the private sector in a range of water modelling and environmental management projects. A formal review and survey of the HowLeaky model has identified the importance of the model for the Queensland Government and a range of stakeholders across Queensland and recommended developing a formal management structure for its governance.

As identified in a formal review of the model by Pembleton (2016), the value of HowLeaky for stakeholders is significant. This is due to reasons such as model simplicity for application, computational efficiencies, application of the model in Great Barrier Reef (GBR) projects, established community of users, and application of the model in scientific research by universities and researchers. The model is a useful tool for research and has appeared in more than 60 publications (for both PERFECT and HowLeaky), of which 50% are peer-reviewed journal and conference papers. The citation counts for these papers

indicate a considerable contribution to the body of scientific knowledge around soil water management (Pembleton, 2016), despite the limited support and investment the model has received in the last 20 years. Also, HowLeaky has been used in a range of technical reports which have been difficult to fully identify. Thus there might be a likely underestimation in application of HowLeaky due to its likely use in internal technical reports which are difficult to access/document.

Before commencement of this project, the formal review of HowLeaky (Pembleton, 2016) had identified issues with model governance and potentials for significant decline in the model's performance over time without a formal governance and development framework. A loss of the HowLeaky model could result in delays to Queensland Government and private sector projects, and research with an increase in the cost of the delivery of the projects in which the model has been used previously. This is due to lower efficiency and lack of a pesticide model within likely alternative models e.g. APSIM.

Now concerns about model governance has been addressed by redevelopment of the HowLeaky model in this project. The new HowLeaky will continue to address the requirements of the model users with a greater focus on facilitating model development, dataset management, and model governance. The model now has a formal peer review structure for development and governance to ensure HowLeaky continues to be a useful tool for water modelling. The project developed facilities to build a command line version of HowLeaky for large scale simulations as was agreed in the project contract. This executive single file simulator is available for download and will be updated with developments approved by the panel. Although simulation run and visualisation of the results were not part of the project deliverables, they have been included in the website at a basic level. Table 2 provides a comparison of the new and old versions of HowLeaky and also with APSIM as the main alternative model.

Table 2 A comparison of the new HowLeaky with the old version and also with APSIM as an alternative model.

Item	New HowLeaky	Old HowLeaky	APSIM
Owner	Queensland Government	Not clear	Multiple organisations
Science: Peer review publications	Not yet (too early)	About 30 peer-reviewed publications	Many publications, much more than old HowLeaky
User interface	Basic functionality for simulations Comprehensive for model development	Comprehensive for simulation. No interface or facility for development.	Comprehensive for simulation.
Formal structure for model development?	Yes	No	Yes
Dataset management facility?	Yes	n/a	n/a
IP & licence	Operates under the Creative Commons Attribution-ShareAlike 4.0 International License. No restriction	n/a	For public release but limitations for commercial use.
Governance and peer review	Established	n/a	By multiple organisations and more complex than new HowLeaky
Sustainability & maintenance	Sustained by hosting in USQ	No support	Multiple organisations
Exposed to user reaction	Not yet	Yes	Yes
Model tested?	Not yet	Yes	Yes

3 Re-development

The activities for re-development included migrating the code into different programming language and developing a hosting and governance framework. This included the translation of the code from C++ to C# and developing a web-based platform for re-development, the establishment of a peer review process, the validation of changes in the code, dataset management, and simulations on the cloud.

3.1 Source codes

The original C++ version of HowLeaky (old version) was written using Borland/Embarcadero C++Builder using a range of open-source and commercial third-

party libraries. Many of these libraries were deprecated over 10 years ago, and the original source-code can no longer be compiled using the latest compilers. In addition, the code was poorly structured using techniques originating 20 years ago. Therefore a rewrite of the code was required, and it has since been rewritten in C# using the .Net Core framework. While the original C++ code compiles into an executable that runs quicker than the equivalent C# code, the benefits of developing in C# are much greater. This includes:

- Financially efficient development and maintenance.
- Larger development community.
- More choice of open-source libraries.
- C# is the most popular technology for climate/land-use modellers in Australia.

The new coding language C# provided with a modern and simplified coding environment to facilitate development processes for the model developers.

The simplicity of C# provides a great benefit for developers. In reality, the C# -based executable only runs marginally slower than the C++ equivalent, and much of this is to do with updating the UI.

The new HowLeaky code in C# uses the Entity Framework (Lerman, 2010) and developed within Microsoft Visual Studio. The object-oriented programming was employed and the system used reflection to interrogate structures and properties of the classes where required. Decorators were used to describe the parameters and units of measures for the properties of Input Model objects and outputs from Controller objects. This made classes and the objects self-describing using reflection, a feature that was also used to develop pages in the User Interface.

3.2 Structure of the code

The model consists of a base code for simulation of the soil-water balance and a range of optional sub-models for simulation of soil erosion, N and P transport, irrigation, and herbicide/pesticide for different types of cropping and management practices. This structure is illustrated in Figure 1 from the technical manual of HowLeaky (Queensland Government, 2019) to show the base model, optional sub-models, inputs and outputs. The inputs are grouped and can be different for each sub-model; they are laced in separate XML files for climate, soil, crop, irrigation, nitrate, phosphorus, tillage, pesticide, and solutes. The structure is highly modular and configurable to maximise computational performance by avoiding redundant calculations (Queensland Government, 2019). The outputs are daily/monthly/yearly time-series which can be set up by the user.

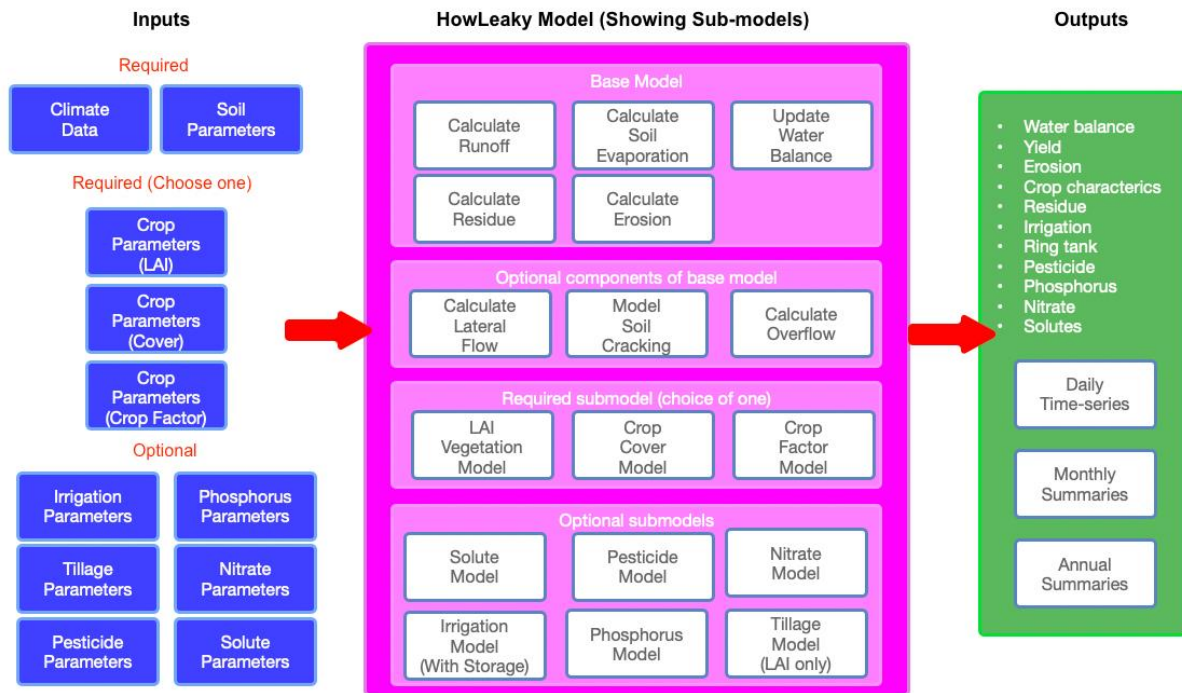


Figure 1 Structure of HowLeaky showing main model, sub-models, inputs and outputs (Queensland Government, 2019).

The operation for “daily simulation” is the central routine of the engine. These operations occur each day while the simulations are grouped into sub-models. All the processes from the start date to the end date are controlled by a “controller” of each submodel as are shown in the Figure 2. The daily simulations first update variables of the core model of daily water balance before running sub-models that are waiting to be simulated. At the end of each day, all variables of the base model and sub-models are updated.

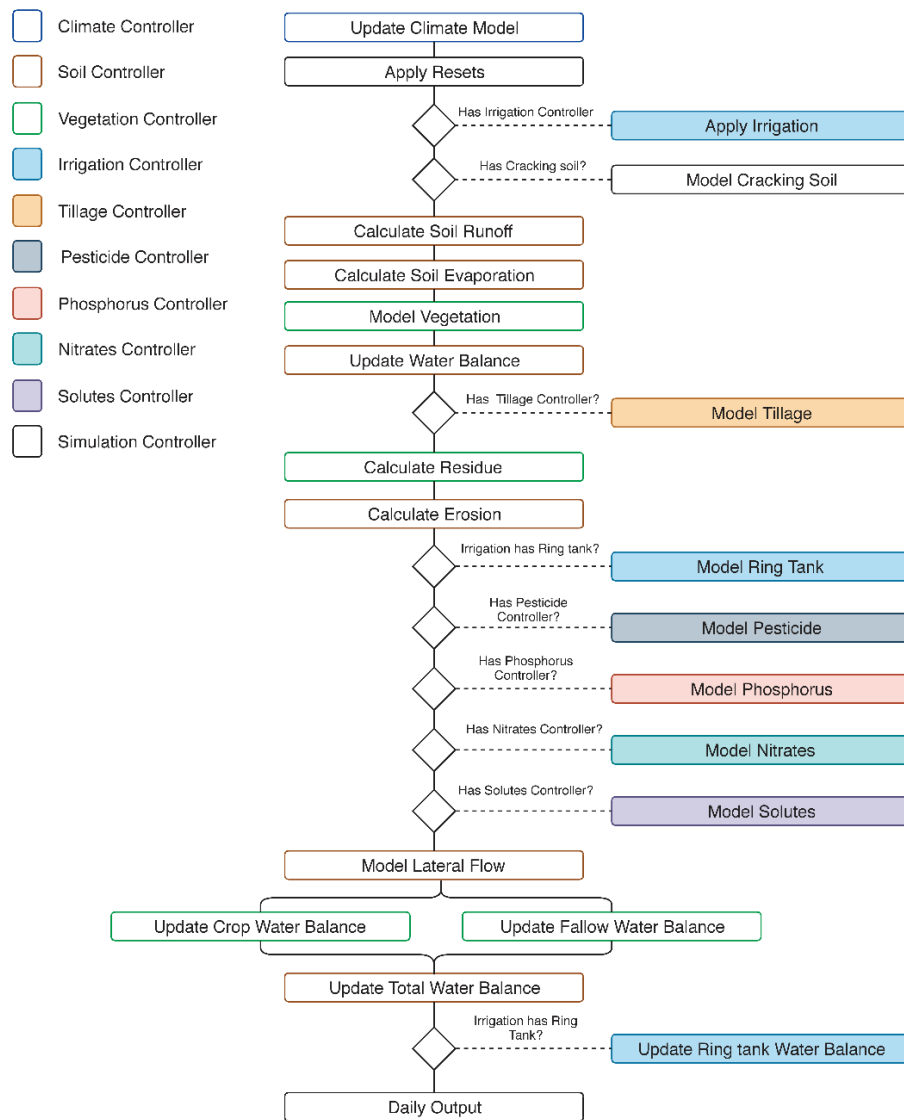


Figure 2 Structure of daily simulation flow in HowLeaky.

3.3 Hosting infrastructure

HowLeaky is hosted on Nectar Virtual Servers running on the QRIScloud availability zones. The project developed three servers on Nectar infrastructure which were used during the project for developments. They are also applied after completing the project to run the system.

These servers are:

Data Server: As a MySQL server and data storage place during development and as the MySQL Database server after development was completed.

Build/Run: For the re-development of HowLeaky, running the web application, and managing administration activities (HowLeaky Panel). This server was used for build during the project and as a run server during and after completion of the project.

Web Application server: To host a web application to interact with the users. The expectation is to have user numbers of >50 after the project is completed. While the web server is interacting with users, the simulation runs occur in the Build/Run server. The Build server automates the building of the HowLeaky Engine (the simulator) using Jenkins. This has hooks into the GitHub repository and builds are triggered by user activity. The database services are provided by MariaDB (fork of MySQL) on the data server. Access to the data is provided via API. The website and API are hosted in Docker containers on the Web server. There are two containers (production and staging) running on the Web server. The staging server checks for source code changes every 10 minutes. The production server checks for changes each midnight. If source code changes are detected, the servers rebuild the website and API and then restart those services. The Docker containers allow for easy deployment of feature testing sites if this required.

More details of hosting infrastructure are explained in Appendix 1.

GitHub

GitHub is a web-based hosting service for source code management including version control. GitHub is used as a public repository host, an important part of the build and automation of the code and dataset submission to the Panel for review. HowLeaky source code in GitHub can be accessed in the following link (Figure 3)

<https://GitHub.com/HowLeaky/HowLeaky.git>

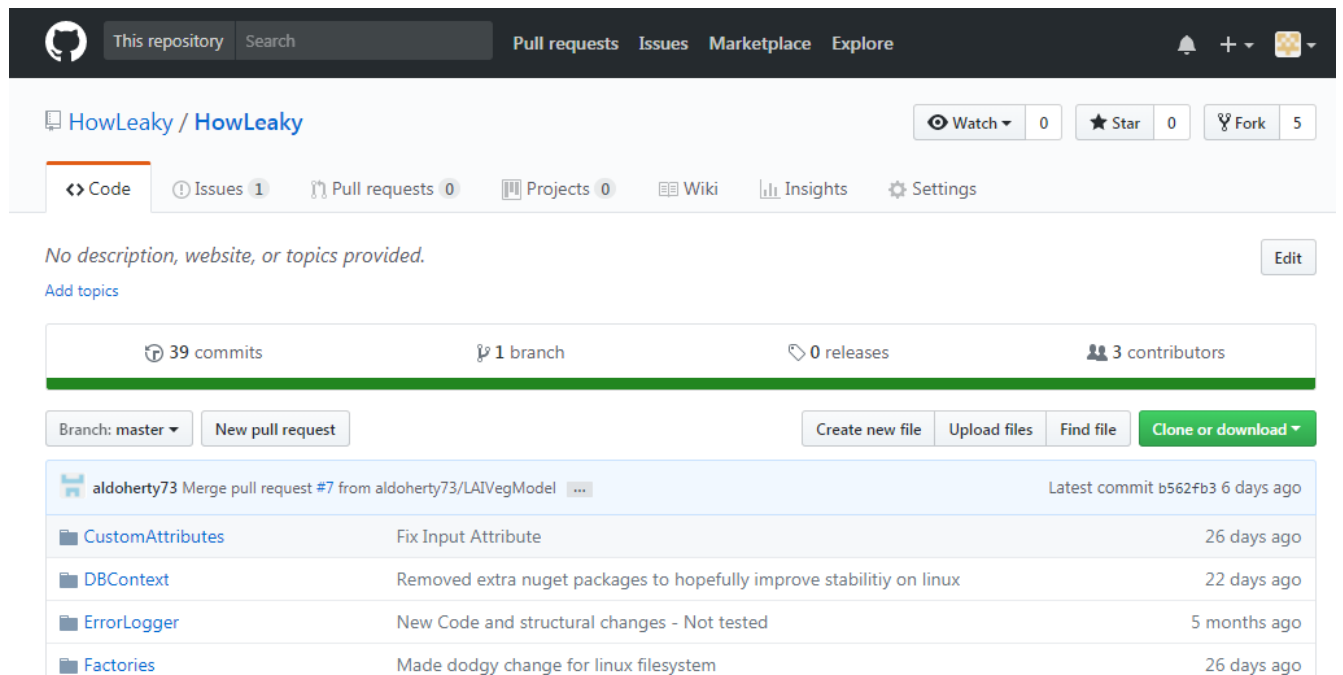


Figure 3 The public HowLeaky code repository in GitHub.

<https://GitHub.com/HowLeaky/HowLeaky.git>

University of Southern Queensland | HowLeaky Development and Governance

A HowLeaky developer can use “forking” for producing a personal copy of the source code and bridging it to the original repository. An instruction of the process for “forking” is provided in Appendix 2. To trigger a build of the model, developers undertake a “Pull” request from the GitHub repository. This triggers Jenkins to undertake a model build. Further details explaining a pull request are provided in Appendix 2.

Jenkins

Jenkins is an open source automation server used in this project to automate the build and review system. It supports GitHub and is triggered by a pull request in GitHub. The Jenkins is used for building the application i.e. the executable HowLeaky file. It can build an executable for both testing (by model developers) and build the official versions approved by the HowLeaky model review panel. The Jenkins page for HowLeaky is shown in Figure 4. The Jenkins test and build are also triggering codes for producing the validation reports and placing them in a workspace accessible to the developers and the review panel. The workflow for the test and build framework is explained with details in Appendix 3.

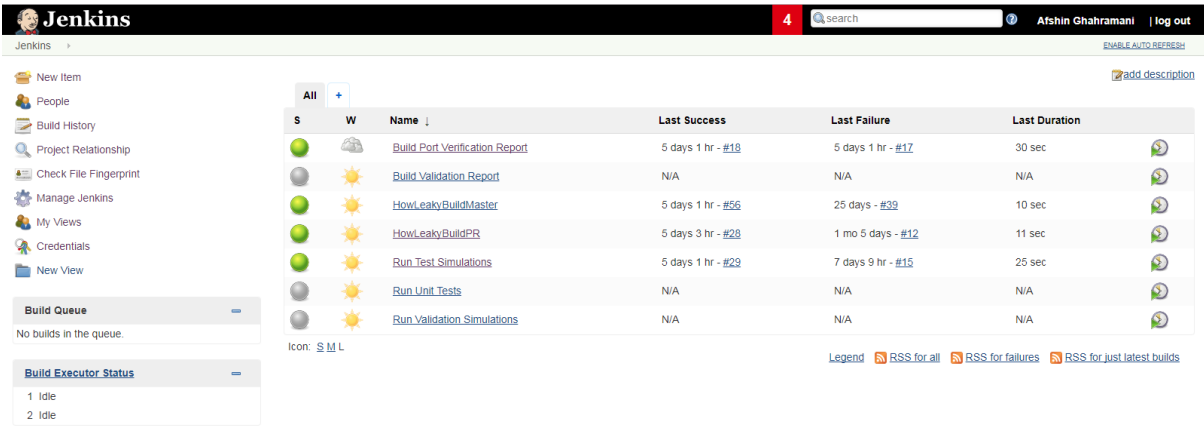


Figure 4 The Jenkins page of the HowLeaky project.

4 Application programming interfaces (APIs)

An API was developed for the HowLeaky platform using Microsoft .NET Code Entity Framework. It provides full functionality for retrieving individual projects' simulations and models, allowing them to be edited in the User interface (UI) and then transported back to the database for storage. The API also has the functionality to run simulations and generate simulation reports. The HowLeaky API is based on Microsoft .Net Core Framework technology. The API provides a platform to rapidly develop more targeted web and mobile applications using HowLeaky as a modelling engine. Through the API, simulations and models can be parameterised and run then summarised for the custom report through targeted applications.

The HowLeaky API also uses Swagger (<http://howleaky.com/swagger/index.html>) interface to expose the endpoints in a user-friendly way for testing purposes and also provides a 'familiar' platform for describing the endpoints and the data structures. The platform is secured using a token with a 'Bearer' claim. To obtain a token, a user must first log in to the platform using the *authenticate* endpoint (Figure 5).

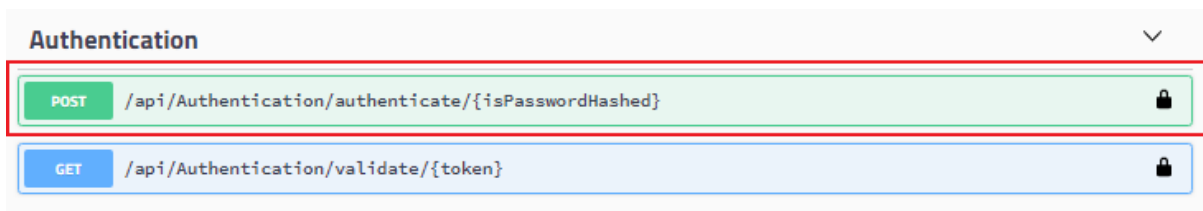


Figure 5 Identifying 'Authenticate' Swagger endpoint.

The user should then enter their credentials (username / password) in the 'userCreds' object and execute the query. The resulting object should look like the following image (Figure 6).

The first thing to note when executing this command is the return code, which is shown at the bottom of Figure 6. A return code of 200 indicates the successful reply to the request. The token is shown in the highlighted area and labelled. This will be needed for authenticating a user on the Swagger API. It should also be noted that a curl command is also created, which can be used in a command line terminal or in an automated application.

Authentication

POST /api/Authentication/authenticate/{isPasswordHashed}

Parameters

Try it out

Name	Description
userCreds (body)	<p>Example Value Model</p> <pre>{ "email": " ", "password": " " }</pre> <p>Parameter content type: application/json-patch+json</p>
isPasswordHashed * required boolean (patch)	Default value: false

Responses

Response content type: application/json

Curl

```
curl -X POST "http://localhost:55244/api/Authentication/authenticate/false" -H "accept: application/json" -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bm9udGVmbmFzZSI6IjI1IiwiaWF0IjoiYXNjaWZlbnR5c2V0eS5jb20uY2V5bW4xYjU0Q040e2_g07pX_exqKV40Elnfn9hdQ" -H "Content-Type: application/json-patch+json" -d "{ \"email\": \" \", \"password\": \" \" }
```

Request URL

http://localhost:55244/api/Authentication/authenticate/false

Server response

Code	Details
200	<p>Response body</p> <pre>{ "firstName": "A", "lastName": "P", "email": " ", "isValidated": true, "agreementAccepted": "2019-05-09T02:10:35.990", "password": null, "admin": false, "projects": [], "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bm9udGVmbmFzZSI6IjI1IiwiaWF0IjoiYXNjaWZlbnR5c2V0eS5jb20uY2V5bW4xYjU0Q040e2_g07pX_exqKV40Elnfn9hdQ", "id": 29, "dateCreated": "0001-01-01T00:00:00", "dateModified": "0001-01-01T00:00:00", "dateAccessed": "0001-01-01T00:00:00" }</pre> <p>Response headers</p> <pre>access-control-allow-origin: * content-type: application/json; charset=utf-8 date: Tue, 11 Jun 2019 23:58:27 GMT server: Kestrel transfer-encoding: chunked</pre>

Responses

Code	Description
200	Success

Token

Figure 6 Operation and results from 'Authenticate' Swagger endpoint.

Once a user has a token, they can Authorise the Swagger API for any further transactions. The Authorisation process injects the 'Bearer' token into any further requests made through Swagger. It is important that the word 'Bearer' followed by a space is inserted before the token is pasted in, see Figure 7 and Figure 8.

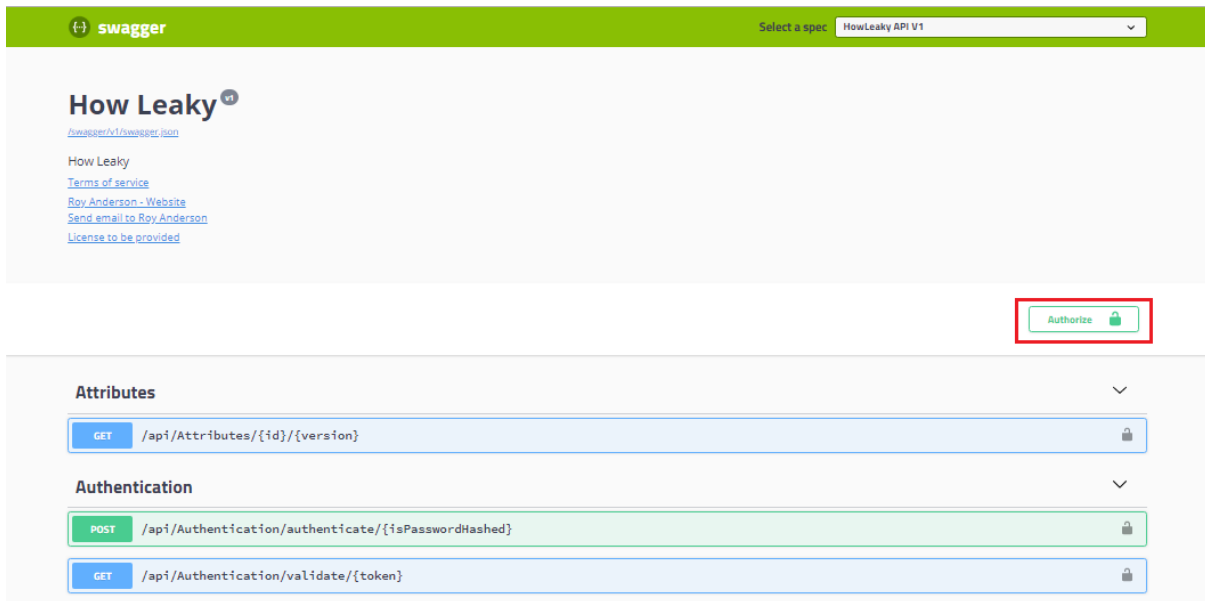


Figure 7 Location of the 'Authorise' button in the HowLeaky Swagger interface.

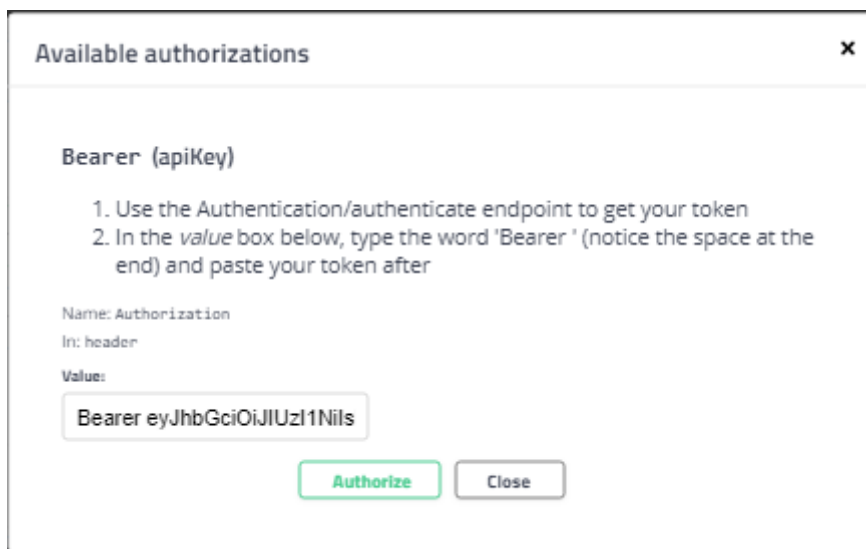


Figure 8 Sample input for Authorisation on the HowLeaky Swagger interface.

4.1 Docker Dev-ops System

A Dev-ops system was created for this project to continuously release versions of the both the API and UI platforms to a staging web site and a release (master) web site (see Figure 9). The dev-ops system relies heavily on both Docker and Git. A docker container is initiated with GitHub credentials and parameterised with a Git branch for the HowLeaky API project and a Git branch for the HowLeaky UI project. A connection string to a specific database is also included in the parameterisation.

The two common containers are located at `www.howleaky.com:80` (as 80 is the default HTTP port, there is no need to explicitly include it), for the release server and

www.howleaky.com.2700 for the staging server. There are mechanisms inside the containers that monitor merges to the branch(es) on which they have been instantiated and automatically pull and rebuild and the source code when a change has been made, for both the UI and API servers.

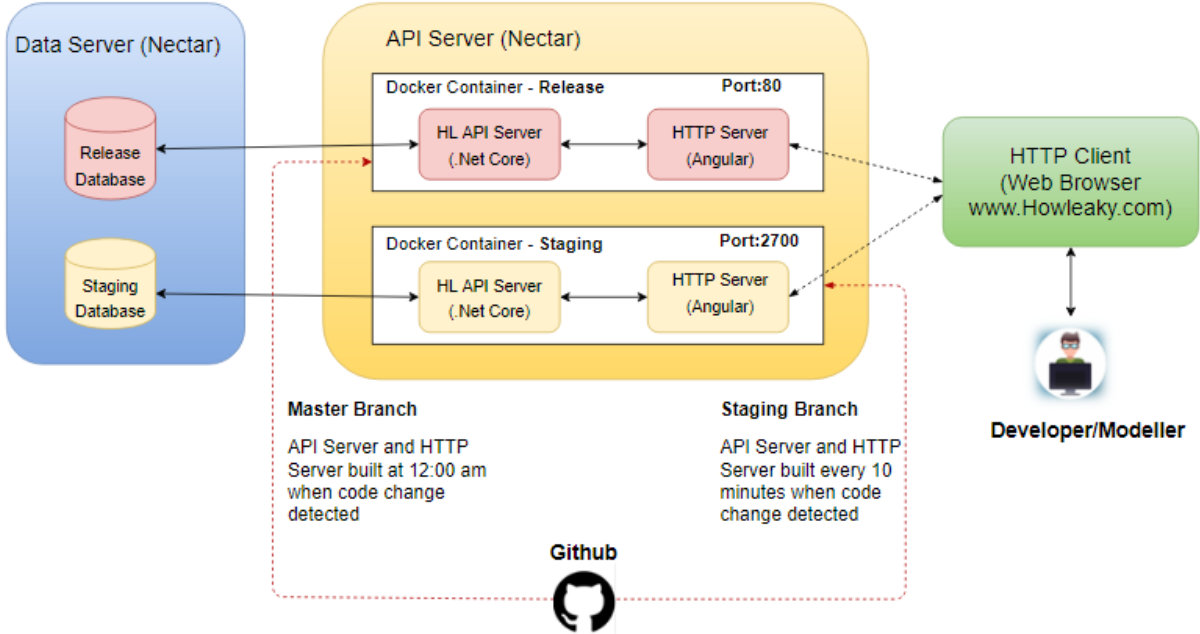


Figure 9 Architecture of the HowLeaky dev-ops system for model development.

Development work is carried out on the staging container so that work can be tested and verified before merging to the respective master branches. The system is not restricted to a staging and master branch. Any feature branch can be rolled out quickly for independent testing prior to merging with either staging or master branches. The master branch is tested for changes at midnight (AEST) every night and all other branches are checked every 10 minutes for changes.

This system requires virtually no manual interaction to update servers with new code to test new features as all the control flow is effectively handled by Git merges.

5 Code development and auto-validation

As was explained in the previous sections, the source codes of the new version of HowLeaky is hosted on GitHub <https://GitHub.com/HowLeaky/HowLeaky.git>. GitHub is connected to Jenkins for building and submission of the code. A workflow implemented in the project build server to automate the submission is presented in Figure 5. The workflow has been documented on <http://howleaky.com/contribute/developer-workflow>.

Developer tools are available to streamline the process of development and code submission to the panel and recommendations are made in the HowLeaky Website. After a developer has made a 'Pull Request', the panel will have access to the interface on GitHub to manage the acceptance or rejection of the proposed change. This GitHub interaction will include:

- A mailing list to the panel and other HowLeaky developers.
- Automated email notification to Panel members when a Pull Request has been made.
- Interactions with developers and an automatic trigger for the Build and Test Framework.

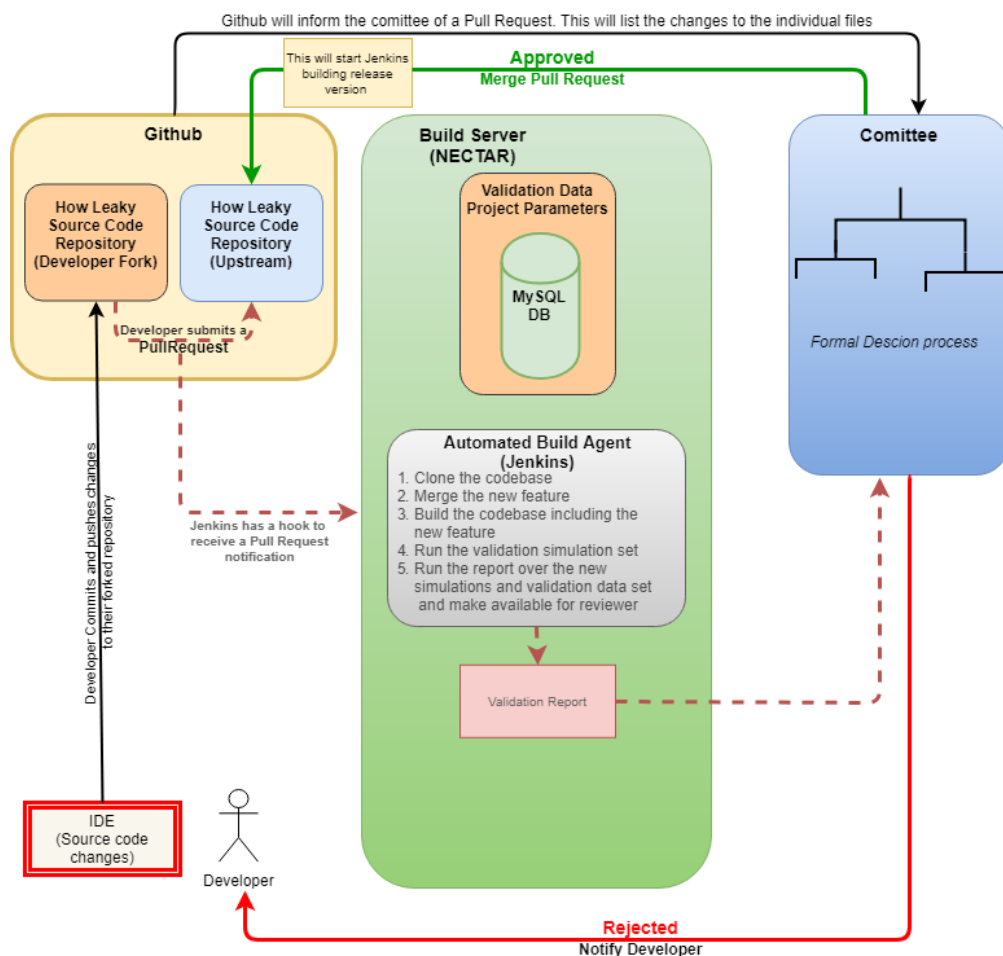


Figure 10 The process of development and code submission.

- The HowLeaky build and validation server is a Jenkins (<https://jenkins.io/>) automation server located at <http://www.howleakybuild.net.au/>. Jenkins can also be accessed via the new website.
- It is set up to run test and validations over the key datasets when a pull request is made via GitHub.
- It interacts with GitHub and can inform the Panel and the developer when source code pull request fails to build or fails any of several tests.

6 HowLeaky Panel

Objective

HowLeaky re-development and use will be managed by the HowLeaky Panel (Panel) originally appointed by the Department of Environment and Science and Department of Natural Resources, Mines and Energy of Queensland Government and the Centre for Sustainable Agricultural Systems, the University of Southern Queensland. The activities of the Panel is overseen by the Steering Committee which consists of the delegate(s) from the Queensland Government and the University of Southern Queensland. The Steering Committee ensures the existence of on-going support for the development and use of HowLeaky.

Roles

The Panel members are individual stakeholders who act to advise the Queensland Government and third parties on technical and scientific matters relating to the model, act as the custodian of the source code and core datasets including:

- (a) Source code change and development including release management, modification and development of the source code and science,
- (b) Acceptance or rejection of the improvements in the source code proposed by developers within the following categories:
 - Minor improvements: these are small changes in the current code, cosmetic changes, and bug fixes with no effect on model verification or improvements. The minor changes need to be accepted by at least one of the Panel members.
 - Major improvements: these include e.g. new science, changes in formulations, the addition of new modules. The major improvements are accepted after agreement of the majority of the Panel members and review of the new validation.
- (c) Legal, financial, scientific issues and relationships with developers,
- (d) Planning and seeking support for the further development of HowLeaky.

Membership

The Panel was established via the HowLeaky re-development project under USQ-DES Collaborative Research Agreement (2017-2019). The original members of the Panel were proposed by the project leader and by the Steering Committee and agreements were formalised during establishment of the Panel. From time to time, the Panel may determine the number of members of the Panel and add a new member or remove existing members in consultation with a current Panel member. New members can be HowLeaky user/developers from around the world who can apply through the HowLeaky

website to become a member. A Panel member may resign at any time by notifying the Panel Chair in writing. The structured relationships between HowLeaky stakeholders is presented in Figure 6.

Activities of the Panel

The Panel may meet and regulate its meetings to fulfil its obligations.

The Panel Chair manages the Panel activities to support the Steering Committee. This will include:

- (a) Preparing materials and presenting to the Steering Committee (Materials may include meeting agendas, annual reports, and project plans),
- (b) Facilitating Panel activities with the HowLeaky community and Steering Committee,
- (c) Facilitating third party involvement? in the further development of HowLeaky source code and science by providing advice where appropriate,
- (d) The direction of the Panel will be determined by the Panel members and will be functional after the approval of the Steering Committee.

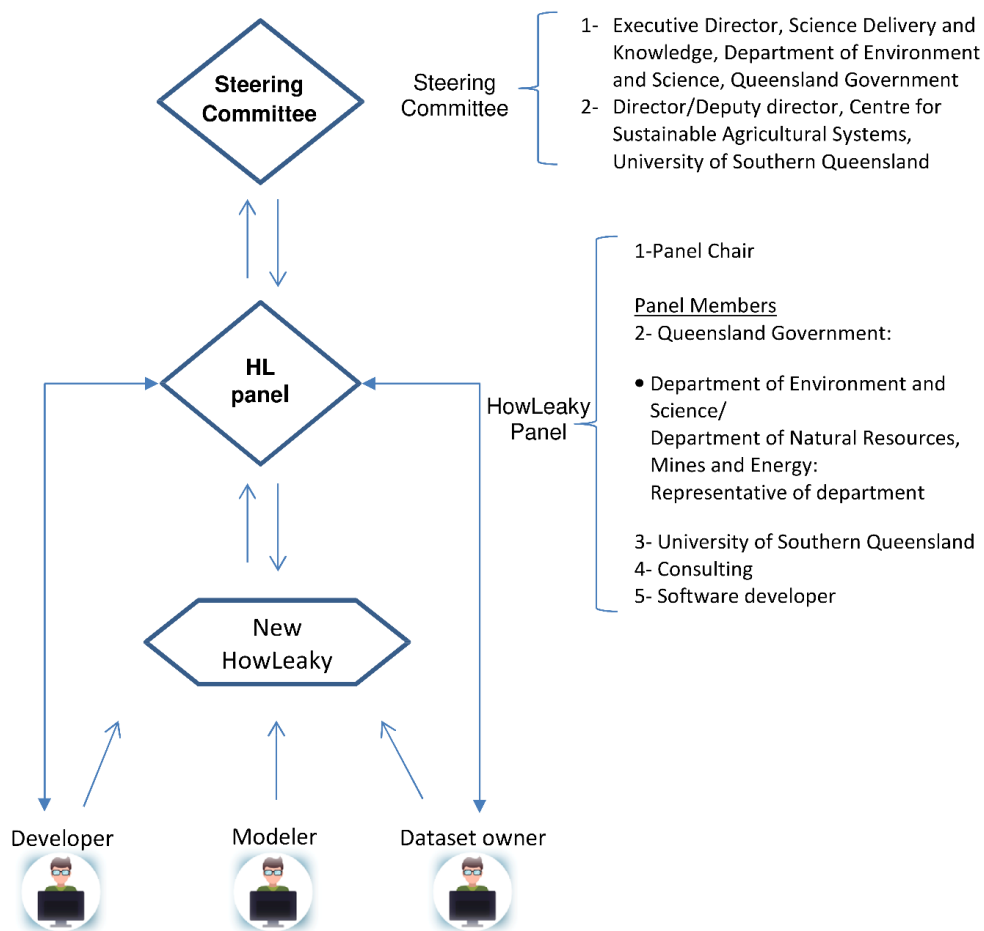


Figure 11 Interaction between HowLeaky stakeholders (at the time of initiating original Panel).

After a developer has made a 'Pull Request', the Panel will have access to the interface on GitHub to manage the acceptance or rejection of the proposed change. This GitHub interaction will include:

- A mailing list to the Panel and other HowLeaky developers.
- Automated email notification to Panel members when a Pull Request has been made.
- A method to easily interact with developers and automatically trigger the Build and Test Framework.

7 Code change submission and automated review system

The new version of HowLeaky has been placed in a GitHub repository, as explained in the earlier sections, GitHub is a cloud-based source used for storing the code, build, and automation framework. GitHub is connected to Jenkins, a build and test framework located on the project build server (housed on Nectarcloud). A workflow was designed and implemented in Jenkins and GitHub to automate the development processes of HowLeaky. This process is outlined in Figure 7. At the time of this report, validation reports are being automatically generated for verification testing through an automated review process.

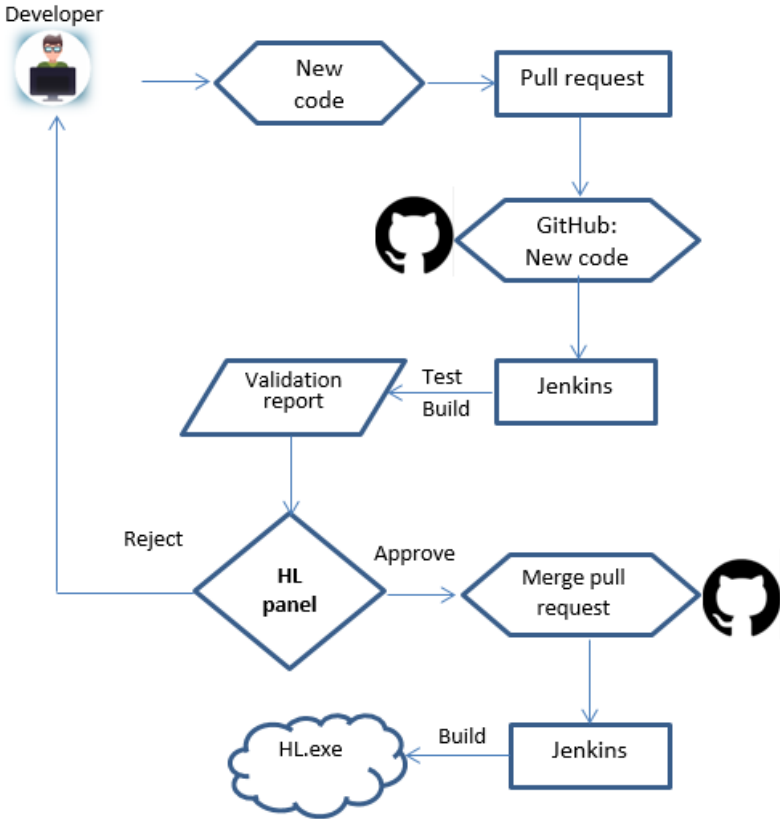


Figure 12 The processes of a code submission and automated review for HowLeaky.

8 Dataset management

Datasets associated with models and parameters are becoming increasingly important and can be as the primary intellectual component of a research platform (Dekker, 2006). Archiving and using datasets are vital for a transparent model re-development verifications process once codes are altered for developing a better model. They can also be valuable data (in combination with parameters) available to the public for future research. These datasets are an essential part of the HowLeaky model to be considered as a platform with a scientific infrastructure. Datasets on the HowLeaky website have been gathered over time by the public sector and recently there is a shift in public culture to make government data more accessible and usable by public e.g. research and consultancy as a priority of the Australian Government (Paterson et al., 2016).

8.1 Data and parameters archived within the hosting framework

In total, there are 150 datasets in the database and all available data of each datasets are accessible to users via the website; those with observed field data are used for verification of the HowLeaky model once code is being re-developed (Table 1). These datasets are for studies across Australia and overseas but those with observed data to be used for model validations are in Queensland (Table 1). Geographic distribution of datasets across Australian states and territories is shown in Figure 13.

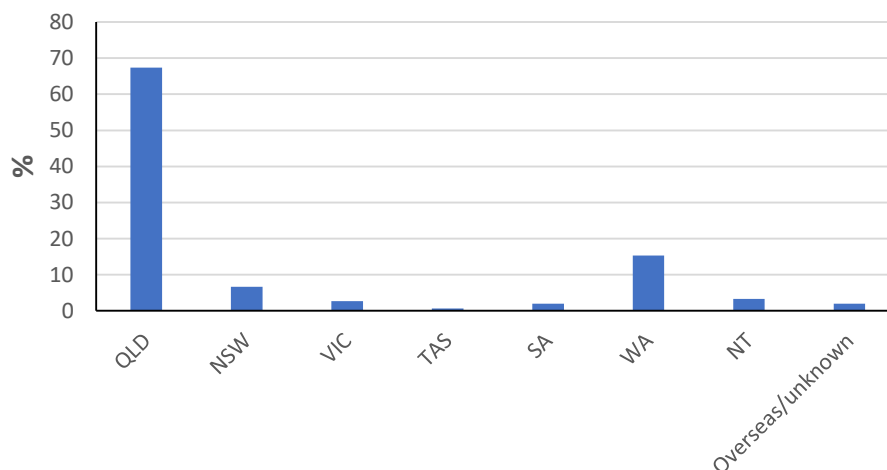


Figure 13 Distribution of datasets across Australia.

Almost all available experimental datasets were already collected into a Joomla MySQL Database of the old HowLeaky website. The Project arranged for access to this database and ensured automated transfer of all the datasets to the new server using a developed C# code. Data manipulations was performed to fit the data to the new MYSQL database.

The MySQL database developed through this process has been placed on the data server. The MySQL database can be accessed using (not for circulation):

Username: public

Password: provided upon request

Server IP: 203.101.224.184

No user interface has been developed for the database in the above IP as this was not a requirement of the project (contract). A MySQL client can be used to build a connection to the database instead. Any future addition of data sets to either users' own projects or addition to validation projects will be done via the web application UI.

All the datasets are accessible on the website as are explained in the section 8. All the details of the datasets are presented at four levels in the "DataSet" section of the website.

These four levels are

1. General: Purpose, method, key findings, location and related studies
2. Experiments: What, when, where, how, project admin and key references and sources
3. Observations: Experiment details
4. Key Figures: Key findings, tables & figures.

The dataset for each site can be chosen from the map or a drop down list beside the map.

8.2 Key data sets for model validation

Experimental datasets are used for validation of the HowLeaky model after code re-development. The process of the mode validation is automated and the procedure is explained in this report and the website. At the time of writing this report (June 2019), there were eight core experimental sites with observed field data of rainfall, cover, runoff, sediment, and plant available water, phosphorous, dissolved inorganic nitrogen , chemicals i.e. 2,4-D and Atrazine (Table 3). Datasets include:

- 1- Observed data
- 2- Model parameters that are calibrated for each site using observed data
- 3- HowLeaky simulation model of each site (*.hlk)
- 4- Model assessment report for each experimental site.

Any changes in the original experimental data including simulation files and parameters are subject to approval by the Panel (details are provided below).

Datasets used by third parties for the further development of the model are stored in the HowLeaky database under their own user accounts. These data may be added to the

verification set after being approved by the Panel. Developers may limit public access to their datasets. However, the Panel has the freedom to utilise them for HowLeaky model development. Future updates of the current experimental data and addition of new experimental sites will be through the Panel review processes. The addition of any new experimental dataset will require submission of all four required items by dataset supplier outlined above.

Experimental site:
Brigalow, Capella, Gatton CTF, Greenmount, Greenwood, Kairi, Mt Mort, Wallumbilla.

Update of these experimental data and the addition of new experimental sites will be through the HowLeaky website and approval by the Panel. Custodian/provider of the new experimental dataset will provide all four required items outlined in section 8.2.

After collating key validation datasets and organising them in the database, HowLeaky simulation models of each site (*.hlk) were built/re-built, tested, and placed into the new HowLeaky database. The data is in two forms:

- A MySQL database
- An export file structure for use by the HowLeaky executable.

Table 3 Description of core datasets for verification HowLeaky code's re-development. AAR: average annual rainfall. PAWC: Plant available water capacity. Datasets are accessible at www.howleaky.com.

Dataset Name	Site description	Dataset feature	Available data	Data period	Reference
Kairi Research Station	AAR 1230mm, Red Ferrosol (PAWC 143 mm). Slope 6%, L=20 m, 12 bounded plots (20x5m): bare soil, peanuts, maize, and pasture. Conventional & reduced tillage.	Semi-arid tropics, nitrogen and phosphorous data	Cover, rainfall, runoff, suspended sediment, total nitrogen and total phosphorus	1998-2001	Cogle et al. 2011
Capella	AAR 535mm, Black Vertosol (PAWC 177 mm). Slope ~2%, L=130m. Nine contour bay catchments: Zero, reduced conventional tillage, wheat, sorghum, and sunflower.	5 years data, shallow soil, cropping, central highlands of Qld.	Cover, rainfall, runoff, suspended sediment.	1984-1989	Carroll et al. 1997
Brigalow Research Station	AAR 655 mm, Black, Grey Vertosol (PAWC 138 mm). Slope 3%, L=100m, Four ~13 ha catchments: Brigalow scrub, buffelgrass pasture, opportunity cropping, and legume pasture.	Long duration (30 years+), three vegetation types, nutrient and pesticides data.	Cover, rainfall, runoff, suspended sediment, phosphorus, dissolved inorganic nitrogen, atrazine, 2,4-D.	1965-current	Cowie et al 2007, Thornton et al. 2007; Thornton and Elledge 2013
Wallumbilla	AAR 570 mm, Brown Vertosol (PAWC 188 mm). Slope ~2%, L=100m, Four contour bay catchments (4-6 ha).	17 years of data, tillage treatments, and pasture.	Soil water, cover, rainfall, runoff,	1982-2000	Freebairn et al. 2009

Dataset Name	Site description	Dataset feature	Available data	Data period	Reference
	Aggressive and conservative tillage, wheat, pasture.		suspended sediment.		
Greenmount	AAR 793 mm Black Vertosol (PAWC 288 mm). Slope 6% L=60 m. Five contour bay catchments (~1.2ha): winter crop, burnt, incorporated, mulch, no-till fallow, pasture.	17 years of data, tillage treatments, and pasture.	Soil water, cover, rainfall, runoff, soil movement, suspended sediment.	1976-1990	Freebairn and Wockner 1986a
Greenwood	AAR 657 mm, Grey Vertosol (PAWC 243 mm). Slope ~5%, L = 35m. Five contour bay catchments (~0.6ha): winter crop, burnt, incorporated, mulch, no-till summer fallow.	7 years of data, tillage treatments.	Soil water, cover, rainfall, runoff, soil movement, suspended sediment.	1978-1984	Freebairn and Wockner 1986a
Gatton Research Station	Black Vertosol (PAWC 195 mm). Slope 6-8%, L=30 m. Twelve bounded plots (3x3m), four treatments, 3 replicates: Wheeled stubble mulch, Wheeled zero tillage, Controlled traffic stubble mulch, Controlled traffic, zero tillage.	Controlled traffic (with and without compaction) and tillage treatments.	Soil water, cover and rainfall, runoff, detailed soil physical characterisation.	1995-1997	Tullberg et al. 2001; Li et al. 2008
Mt Mort	AAR 768 mm, Sodosol (PAWC 190mm). Slope 12%, L=10m, Bounded plots, pasture at three cover levels: bare, grazed, and excluded.	Pasture study with extremes of management including bare soil.	Soil water, cover, rainfall, runoff, suspended sediment loss.	1993-1999	Silburn, 1994; Rattray et al. 2006

Database

The HowLeaky API platform is based on an underlying structure provided by a MySQL relational database. The database consists of information about HowLeaky datasets originally gathered from the old HowLeaky website (www.howleaky.net). A range of tables were developed to hold required data structures for storage of the other objects used to perform advanced functionality within the HowLeaky API. Figure 14 presents an entity relationship diagram between the HowLeaky database components.

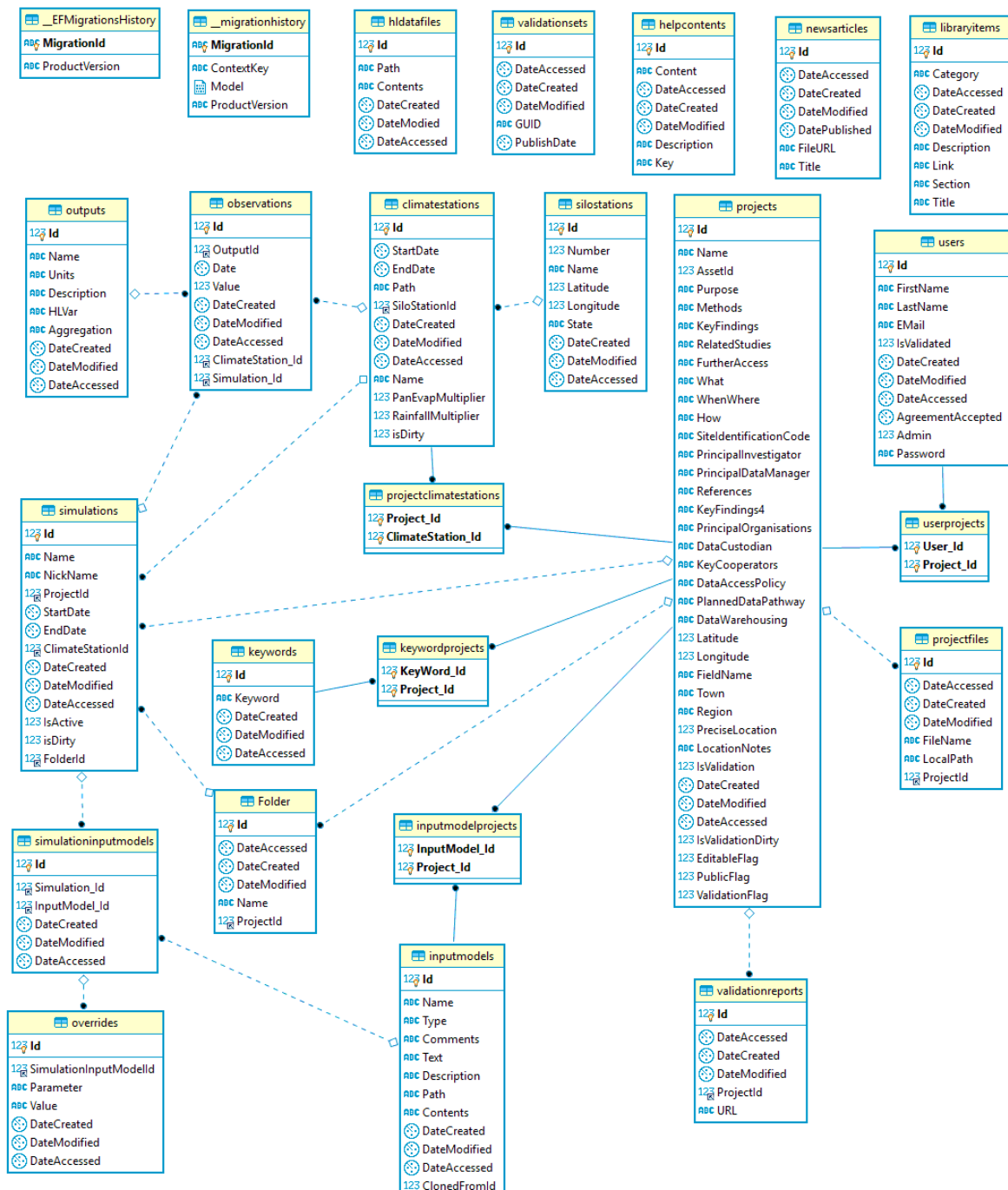


Figure 14 Entity Relationship Diagram of the HowLeaky database.

9 Access to the model parameters

There are a range of the model parameters required to run soil water balance and the optional modules of HowLeaky. The model parameterisation has been performed based on a wide range of empirical studies (Littleboy et al., 1992).

Soil parameters have a physical basis and can be measured in the field or estimated using a range of techniques e.g. pedotransfer functions (Queensland Government, 2019).

Strategic field sampling of the model parameters is time consuming and costly and cannot be conducted on a regular base. The HowLeaky website now has eight core experimental sites that the HowLeaky simulation model (*.hik) has built using field measured parameters and a manual calibration using field observation. These parameter-sets are stored in the cloud (relational database) accessible through the website. These calibrated simulation models are primarily developed for the purpose of validating the model code developments. Their parameters are available for sharing with modellers to promote parameter documentation and encouraging users in further improvements in model parameters as datasets are updated over time. A list of the model parameters and their details are presented in Appendices 1 to 10 of the HowLeaky technical manual (Queensland Government, 2019). The technical manual can be accessed via website: www.howleaky.com/library.

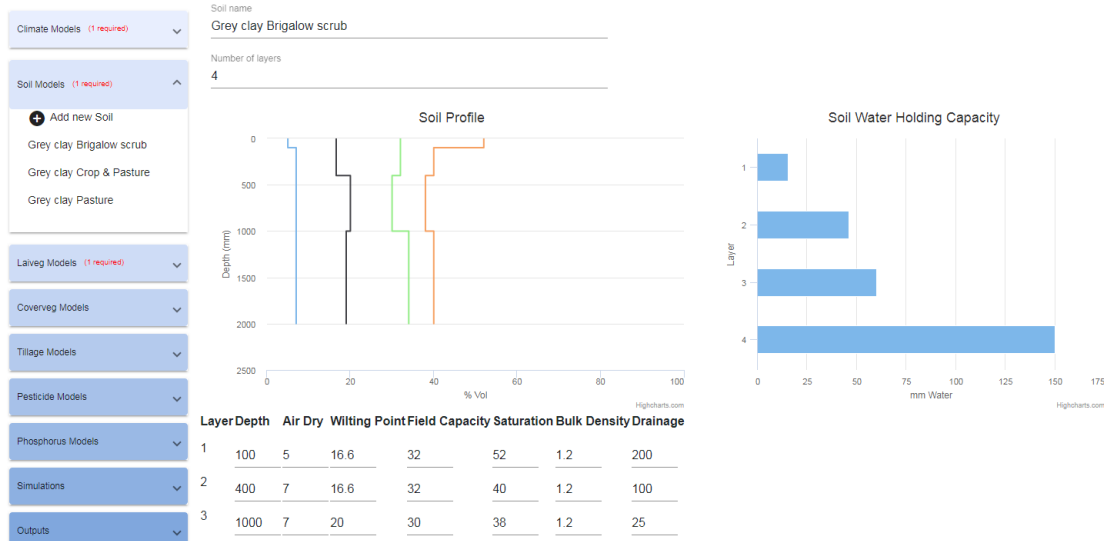
Manual calibration

Manual calibration is generally conducted by an experienced modeller who is familiar with each study site. It is a process to adjust the value of the model parameters manually to match the outputs to the observation visually (Pechlivanidis et al., 2011) or to evaluate performance by using statistical measures, e.g. RMSE. This method is reported to be best suited to experienced modellers (Sorooshian and Gupta, 1995) for HowLeaky, a paddock model, which has a relatively small number of parameters (Ghahramani et al., submitted DNRME and DES, 2019). So many of Howleaky's parameters are measureable that calibration is only requires for a few parameters such as the runoff curve number and two soil evaporation parameters.

A range of calibration software are available for estimation of the model parameters e.g. PEST (Doherty, 2015). In general, these automated methods include an objective function that measures the differences between observation and simulation values and an algorithm to optimise the parameters (Efstratiadis and Koutsoyiannis, 2010). Ghahramani et al. (submitted year?) demonstrated that although an automated calibration helps to reduce error variances, a manual calibration of the HowLeaky simulation is still adequate to estimate soil water balance and water quality across a wide range of land uses and management practices.

The core eight experimental sites and their parameters are accessible on the website under the simulation section. The user can clone these experimental sites, have access to all the parameters and evaluate their different values. The model sensitivities to the parameters can be visualised in the website and the model outputs can be compared to the observed values in the experimental datasets.

Project: Brigalow



Project: Brigalow

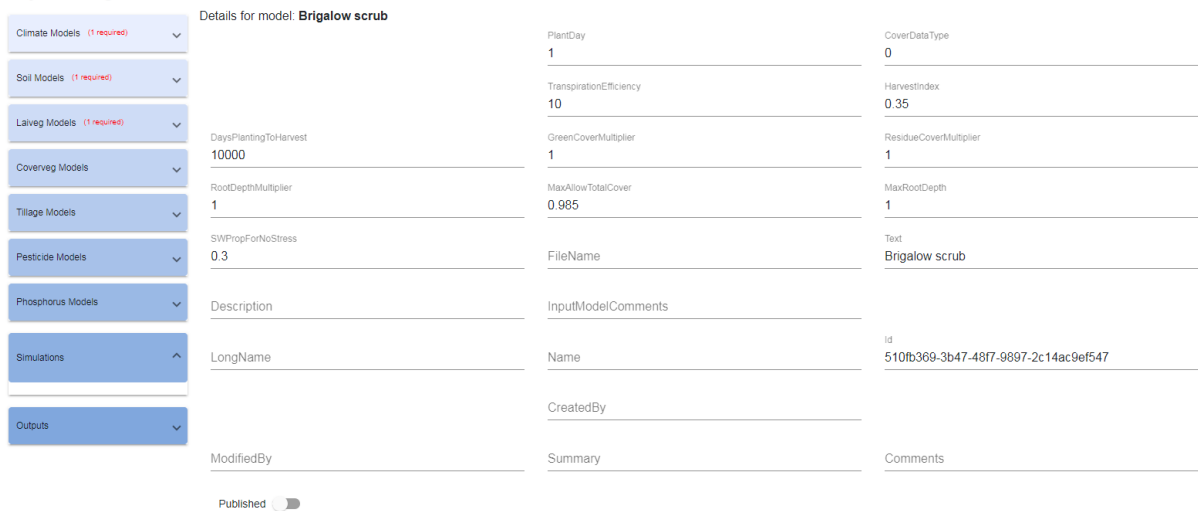


Figure 15 The model parameters of the models datasets are accessible via simulation menu. The parameters of the core validation site are accessible and changes are subject to approval from the Panel.

10 IP & Licence

The licence terms and conditions of HowLeaky software/model/data are described in the License Agreement developed by State of Queensland acting through the Department of Environment and Science. The licence terms and conditions of HowLeaky software/model/data are covered by a Creative Commons attribution share alike version 4 licence <https://creativecommons.org/licenses/by-sa/4.0/>. A clickwrap of active agreement on terms is placed on the new HowLeaky website (hosts experimental data and executive code) and also in the GitHub (hosts source code). The clickwrap licence statement is as follows:

"I have read, understood to the licence terms and conditions [https://creativecommons.org/licenses/by-sa/4.0/] that apply to use of the [Howleaky software/model/data] and agree to be bound by the licence terms and conditions [checkbox to proceed / or "I agree" button to proceed]".

A Creative Commons Attribution-ShareAlike licence provides developers and HowLeaky users with the ability to copy and redistribute the material in any medium or format. They can remix, transform, and build upon the material for any purpose, even commercially.

The licence sets out the following terms for terms:

- **Attribution:** developers and users must give appropriate credit, provide a link to the license, and indicate if changes were made. Developers may do so in any reasonable manner, but not in any way that suggests the licensor endorses them or their use.
- **ShareAlike:** If developers and user remix, transform, or build upon the material, they must distribute their contributions under the same license as the original.
- **No additional restrictions:** developers and users may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

11 Terms of Reference

As part of the project, a Terms of Reference document was developed to clarify the governance structure of HowLeaky (www.howleaky.com/library/HowLeakyTOR_2019.pdf). This document explains the workflow between the HowLeaky Steering Committee, Panel, and end-users; how the panel manages the re-development process in a mainly community-driven manner; management of datasets, model parameters, resource allocations to users; and the addition of new datasets.

12 Future development opportunities

The new platform provides an opportunity for further development of the website and science in the model. Also, there is need to explore the impact of the model on end-users, the implications of the new platform in different projects, and connections to other water modelling tools of the Queensland Water Modelling Network.

12.1 Science development

The performance of the PERFECT and HowLeaky models have been validated with hydrology data for cropping systems in Queensland for runoff, erosion and the movement of nutrients attached to sediments (e.g. Littleboy *et al.*, 1992; Freebairn *et al.*, 2009). The pesticide and phosphorus submodels have been validated by model users e.g. Shaw *et al.*, 2011, Robinson *et al.*, 2011, and are being used by the Queensland Government in the Reef 2050 Water Quality Improvement Plan (<http://www.reefplan.qld.gov.au>).

However, most of the model P2R program validations and parameterisations have not been published in peer-reviewed journals.

An extensive analysis by Ghahramani *et al.* (2019) shows that relying on existing and relatively simple models, e.g. HowLeaky, can be sufficient for decision making, at least for the purpose of estimating water quality signatures for various management options, e.g. in the GBR catchments. Dealing with complex environments and reducing uncertainties may require further parameterisations of the model for different land uses e.g. modelling nitrogen for sugar cane systems and push to a particular approach suited to the model i.e. a more complex parameter calibration procedure. The new HowLeaky platform provides opportunities to enhance the model through transparent development processes. However, the new model requires a longer term comprehensive test by the users.

HowLeaky has not received a substantial investment from the Queensland Government to support further development since the translation of the PERFECT model into HowLeaky in 2003. It has however had limited investment from the Queensland Government and private sector, which helped HowLeaky to survive over the last two decades. This indicated a continuous requirement for the HowLeaky model development which may be accelerated by developing the new platform that facilitates the process. Analysis of the HowLeaky model by Ghahramani et al (submitted) reveals that requirements for further developments or complex modelling exist when complexities are unavoidable, e.g. mixed farming systems (e.g. Ghahramani and Bowran, 2018). However, we should consider that greater complexity may only increase the uncertainties related to the use of the models by users with insufficient skills (Jakeman et al., 2006) or incomplete datasets.

Empirical studies are fundamental to ensuring credible analysis of hydrology, sediment, phosphorus, nitrogen and pesticide movement at the paddock scale. However, there are gaps (in data and algorithms), such as for modelling dissolved nitrogen runoff that require model revisions. Ideally, more data dealing with hydrology, soil erosion, sediment loss, and nutrient and pesticide movement will become available to inform the development of more reliable models. Over time, these data can be added to the HowLeaky platform and be used for the model parameterisation and development. The new platform provides an opportunity for transparent and defensible HowLeaky model development.

Example 1 for science development within the HowLeaky model

The project team, in collaboration with the HowLeaky stakeholders, has identified a necessity for a comprehensive dataset and model analysis of nitrogen and phosphorus

data across GBR catchments. By a further enrichment of the nitrogen and phosphorus datasets and model developments, end users will have a better predictive model to simulate nitrogen and phosphorus in runoff and deep drainage delivery from GBR catchments. This will result in an improved decision and policy-making process for the government and all related stakeholders including regional natural resource management groups, land managers, and growers.

Example 2 for science development within the HowLeaky model

HowLeaky uses a simple bucket model to simulate a one dimensional water movement in the soil. Soil water redistribution is estimated using a linear cascading technique adopted from CREAMS (Knisel 1980). HowLeaky has been proven to model runoff generation and soil water at a paddock scale with high confidence (e.g. Freebairn et al., 2018). However, currently the HowLeaky model has a relatively poor to moderate capability for simulation of soil water dynamics and relationships between soil-water-plant elements (Ghahramani et al., 2018). Although numerical based models e.g. HYDRUS (Šimůnek et al., 2006), have a detailed modelling power they are considered to have a poor capability to model structural dynamics because they do not incorporate plants in the simulations (Ghahramani et al., 2018). Considering computational efficiency of the HowLeaky model compared to the alternative models e.g. APSIM (Holzworth et al., 2014) and existence of the modelling capability in HowLeaky (although simple), developing a numerical model similar to e.g. SWIM (Huth et al., 2012) can add capabilities for simulation of soil structural dynamics in soil.

12.2 Further development of the platform

During the development of the new HowLeaky platform, a further range of development requirements were identified beyond the project scope. These are briefly explained below:

- Administration Level Functionality

This feature allows an administrator of HowLeaky to monitor and control:

- user access to the system
- user load on the system through the allocation of resources and quotas
- security and management of user groups
- privileges granted to user groups.

This would likely take the form of a dashboard page which lists individual system users and details on the projects they have registered. Details on storage and CPU quota consumed by the user and their projects and logs of system access and simulations executed would also be provided.

The administrator would also be able to control/update/delete user membership of a security group. Users would be invited to join groups through an email containing an invitation link. Administrators and project owners would be able to send email to members of a group. This would be particularly useful when an administrator or the project owner needs to broadcast details to those interested or participating in the project. *This is a must do component.*

- Factorial Simulations

This would develop a facility for large scale simulations in local machines or for cloud computing in High-Performance Computing (HPC). This would be useful for Queensland Government research projects, and also third-party users i.e. DSS developers who will be connected to the HowLeaky platform via APIs to use the platform as a modelling engine.

- Distributed Simulations

This feature would allow users of HowLeaky to execute (run) simulations in a distributed environment. This could be added to the new HowLeaky website and facilitate factorial simulations on cloud resources. Running HowLeaky on a server imposes a limit on the number of simulations which can be executed due to resource limitations that will be shared with other users. *This is a must do component.*

- Output Visualisation

The new HowLeaky will visualise the simulation results, however, some simulations can generate a significant amount of data. The ability to process/import this data with R codes is proposed given the popularity of this scripting language.

- Further development of the source code

Most of the required development of HowLeaky is planned to be carried out by the research project teams who will use HowLeaky as their modelling tool. However, at the start, there will be need for the Queensland Government as HowLeaky owner, to develop these kinds of projects so as to ensure that HowLeaky meets users' needs. Related validations and publications will ensure this.

Currently, there is an issue with soil nitrogen module of HowLeaky. The current model is overly simplistic and does not capture the dynamics of interactions, the application of APSIM has not been able to help either in particular in wet tropics. This has been discussed in section 13.1.

Appendices

Appendix 1 Hosting infrastructure

All developing and hosting servers are placed on the National eResearch Collaboration Tools and Resources project (Nectar) infrastructures. Details of the allocated resources are:

- Number of cores: 10
- Number of core hours: 43800 (Default for 10 cores)

The system has a medium size database which may grow over time. Total requirement (by adding some reserve capacity to facilitate instance resizing, relaunching and migration) are as following

- 10 VCPUs
- 40 GB RAM
- 5 instances
- 200 GB instances root file system
- 200 GB secondary disk (volume storage)

Three instances were be made:

- Data server: As a MySQL server and data storage place during development and as the MySQL Database server after development is complete.

The software operating on this instance are Linux & MySQL and services of Git Hook, Release DLL Cache (repository of executables to be used), Validation Report Cache. The initial 30GB of storage was sufficient for initial stages of the project. However, 200GB of volume storage is requested to accommodate the expected increase in storage requirements. This instance has 2 VCPUs, 6GB RAM, and a 30GB root disk. This instance is 24/7 available with 1488 core hour p/m (24 x 31 x 2 cores).

- Build/Run server: This server was developed to be used for the re-development of HowLeaky, running the web application, and managing administration activities (HowLeaky reference panel). This instance was used not only for Build during development activities but also as a run serve during and after completion of the project.

The software operating on this instance are Linux & Jenkins & R & C# (.Net Core) & Web services & DLL cache & Apache/nginx. The build instance had limited users. There was no requirement for replication of this instance because snapshotting will be used as the backup strategy. Services in this instance:

- -Git Hook (Listen for Pull Request);

- -Build Agent (Develop): Pull Git, Build, Batch run validation simulations, Prepare validation report.
- -Build Agent (Release): Pull Git, Build, Add DLL to cache, Generate documentation.
- -Release DLL cache
- -Validation Report cache

The build activity occurs at irregular intervals. This instance has 4 VCPUs with 12 GB RAM and 110 GB root storage. This instance does not need a 24/7 availability thus a 600 core hour p/m will be enough.

- Web Application server: To host a web application to interact with the users. There was a limited number of users during the development, but the expectation is to have many more users (>50) after the project is completed. The simulation runs occur in the Build/Run instance.

MYSQL & Web Application Server needs to be available 24/7. There will be no requirement for replication as snapshotting will be used to make backups.

The software operate on this instance are Linux & Apache/nginx & HowLeaky (Web backed) & DLL cache & Mono.

Services in this instance:

- -Web App UI Server
- -Batch run simulations (Web App)
- -Reporting (Web App)

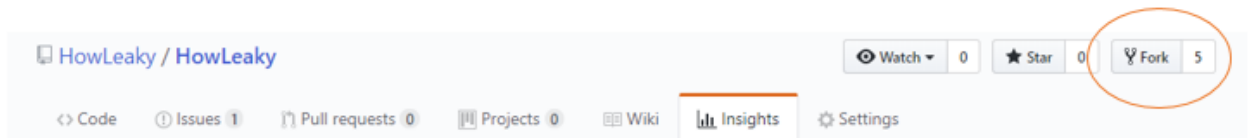
This instance has 2 VCPUs, 6GB RAM, and a 30GB root disk. This instance needs to be 24/7 available with 1488 core hour p/m (24 x 31 x 2 cores) and with an additional allocation for redundancy of 30 core hours p/m.

Appendix 2 GitHub Forking Workflow

GitHub (www.github.com) is a cloud-based Git source control repository host. It is one of the central technologies used in the HowLeaky build and automation framework. GitHub not only stores the source code for the HowLeaky project but also triggers the build agent to build and test any new source code.

To start developing HowLeaky, follow these steps:

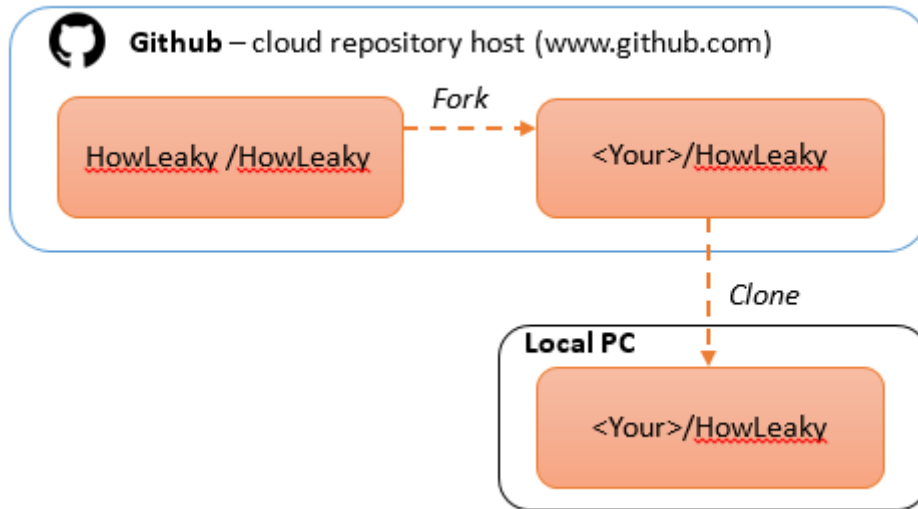
1. You will need to have a GitHub user account
 - Go to www.github.com and follow the instructions to 'Sign Up'
2. After you have logged in, go to this public HowLeaky repository, <https://github.com/HowLeaky/HowLeaky> , and 'fork' it.



This will create an exact duplicate of the main repository in your account. This will be the remote (a Git term means "not local") that you interact with when developing any new features in How Leaky.

3. Now that repository has been forked, you can clone the repository to your local computer. This will create an exact copy of your fork onto your local computer. When you make changes to your local repository, these changes will be 'committed' to your local repository and then 'pushed' to your forked repository.

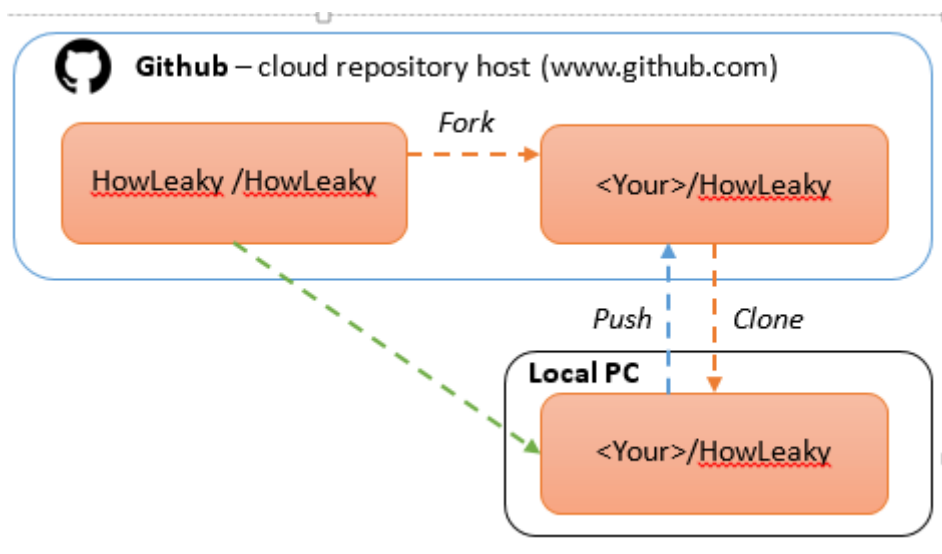
Cloning a repository automatically sets the remote 'origin' repository for the local repository to the fork from which it was cloned. This is important because we are going to add another remote later on.



- When you follow the process outlined above, your repository will contain a 'master' branch. It is recommended that you don't commit changes directly to your 'master' branch but create a new branch for your new feature. This is mainly because we need to keep our 'master' branch the same as the 'upstream' (original) remote repository. Firstly, we need to add the additional repository.

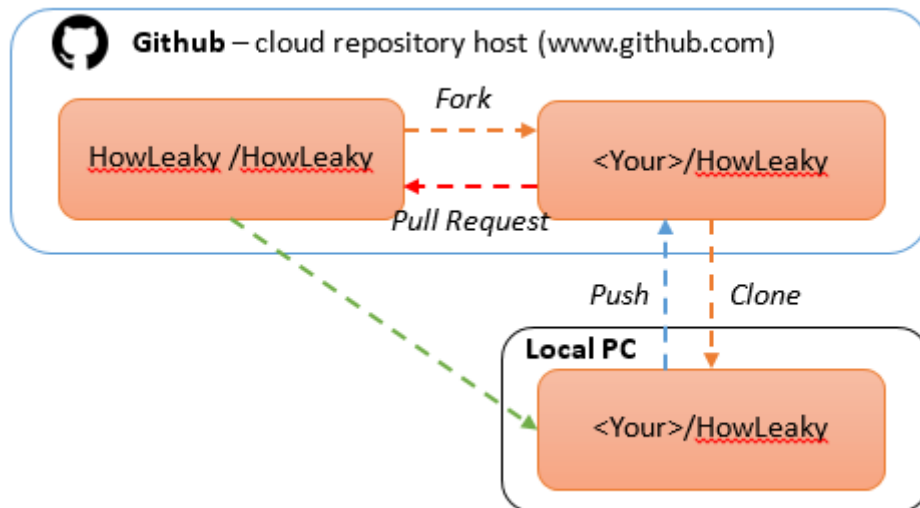
Add a new repository and give it a name, common names used are 'upstream' or 'trunk', the location of the repository is

<https://github.com/HowLeaky/HowLeaky.git> . Now we have this additional repository we can regularly merge any changes made by other developers into our local 'master' branch and then 'push' to our forked remote.



Merge 'upstream' 'master' branch into Local 'master' branch

5. If you have developed a feature that you would like to be added to the HowLeaky code base, you will need to create a 'Pull Request'. A pull request is created via GitHub. Once a pull request is made, depending on the comments, the automatic build and test frame is launched via Jenkins and if required, the Panel will be informed to evaluate the changes.



Merge 'upstream' 'master' branch into Local 'master' branch

Appendix 3: Jenkins Overview

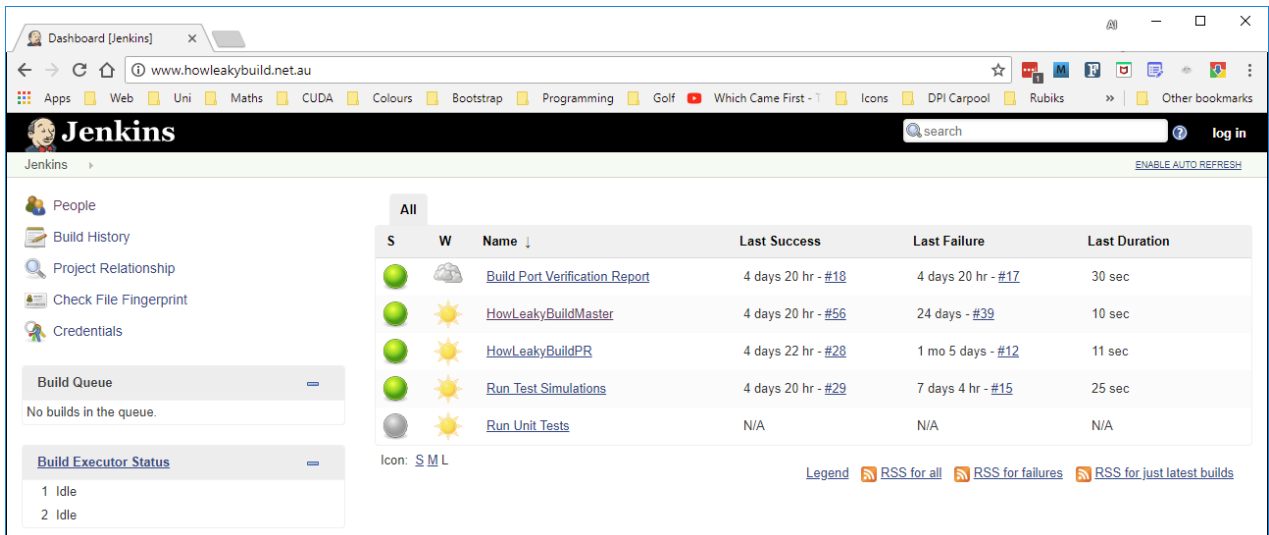
Jenkins is the build and test framework that is being used for the new version of HowLeaky. We have a server located on the project build server at <http://www.howleakybuild.net.au/>.

The site has some public components and some administrative components. The administrative components will not be discussed here.

The public components are the 'workspace' created when the model has been built and tested. The places of interest for most people will be the executables and the reports for the current release version of HowLeaky. These will also eventually be available for download via the project website.

To access this information follow these steps.

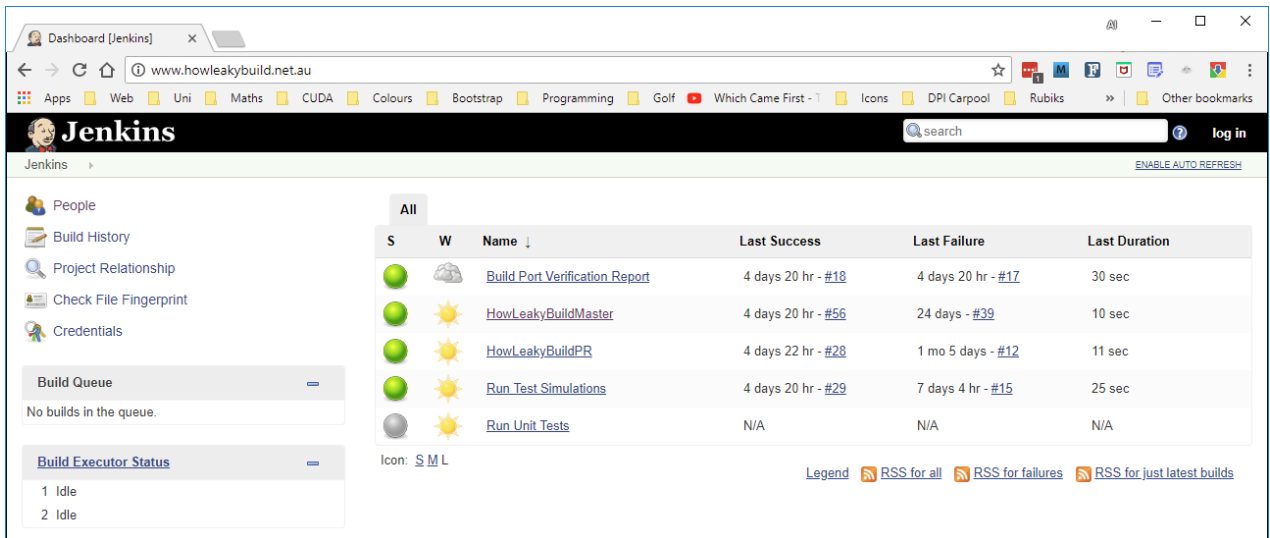
1. Go the URL <http://www.howleakybuild.net.au/>. It should look like this...



The screenshot shows the Jenkins dashboard for the URL www.howleakybuild.net.au. The dashboard includes a search bar, a 'log in' button, and a sidebar with navigation options like 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', and 'Credentials'. The main content area displays a table of build jobs under the 'All' tab. The table has columns for 'S' (Success), 'W' (Warning), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. The jobs listed are 'Build Port Verification Report', 'HowLeakyBuildMaster', 'HowLeakyBuildPPR', 'Run Test Simulations', and 'Run Unit Tests'. Below the table, there are icons for 'S', 'M', and 'L' (Small, Medium, Large) and a legend for RSS feeds: 'RSS for all', 'RSS for failures', and 'RSS for just latest builds'.

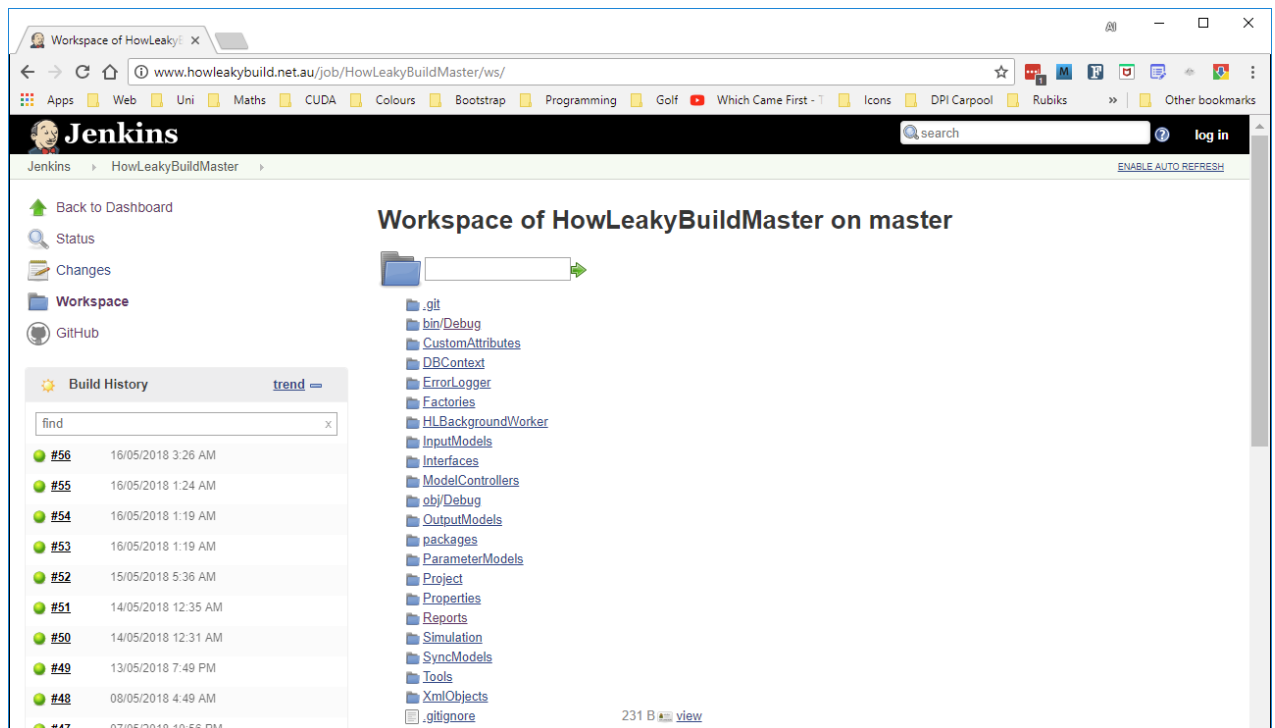
S	W	Name ↓	Last Success	Last Failure	Last Duration
🟢	☁️	Build Port Verification Report	4 days 20 hr - #18	4 days 20 hr - #17	30 sec
🟢	☀️	HowLeakyBuildMaster	4 days 20 hr - #56	24 days - #39	10 sec
🟢	☀️	HowLeakyBuildPPR	4 days 22 hr - #28	1 mo 5 days - #12	11 sec
🟢	☀️	Run Test Simulations	4 days 20 hr - #29	7 days 4 hr - #15	25 sec
🟡	☀️	Run Unit Tests	N/A	N/A	N/A

2. The task that creates the release executables and runs the tests is 'HowLeakyBuildMaster'. This, as the name suggests builds the 'master' branch on the main HowLeaky repository
3. Click the 'HowLeakyBuildMaster' and you will see a page like a screenshot below.



The green dots represent a successful build - if a build is unsuccessful the dots appear red.

- Click on the 'Workspace'. You will see this hierarchy. These are links to directories inside the workspace. The executables are located in 'bin/Debug' and the reports are in the 'Reports' directory. All the rest of the directories and files are the actual source code used to build the model pulled directly from GitHub.



References

- Carroll C., Yu, B., 2018. Queensland Water Modelling Network, Water Model Catalogue: A catalogue of water models used by the Queensland Government. Australian Rivers Institute, Griffith University. Prepared for Queensland Government, Department of Environment and Science.
- Carroll, C, Waters, D, Vardy, S, Silburn, DM, Attard, S, Thorburn, PJ, Davis, AM, Schmidt, M, Wilson, B, Clark, A 2012, 'A paddock to reef monitoring and modelling framework for the Great Barrier Reef: paddock and catchment component', *Marine Pollution Bulletin*, 65, 136–149.
- Doherty, J., 2015. Calibration and uncertainty analysis for complex environmental models. Published by Watermark Numerical Computing, Brisbane, Australia. 227pp. ISBN: 978-0-9943786-0-6.
- Estratiadis, A. and Koutsoyiannis, D., 2010. One decade of multi-objective calibration approaches in hydrological modelling: a review. *Hydrological Sciences Journal–Journal Des Sciences Hydrologiques*, 55(1), pp.58-78.
- Jakeman, A.J., Letcher, R.A. and Norton, J.P., 2006. Ten iterative steps in development and evaluation of environmental models. *Environmental Modelling & Software*, 21(5), pp.602-614.
- Freebairn, D.M., Ghahramani, A., Robinson, J.B. and McClymont, D.J., 2018. A tool for monitoring soil water using modelling, on-farm data, and mobile technology. *Environmental Modelling & Software*, 104, pp.55-63.
- Freebairn, D.M., Wockner, G.H., Hamilton, A.N., Rowland, P., 2009. Impact of soil conditions on hydrology and water quality for a brown clay in the north-eastern cereal zone of Australia. *Aust. J. Soil. Res.*47, 389–402. doi:10.1071/SR07054.
- Freebairn, D.M. and Wockner, G.H., 1986. A study of soil erosion on Vertisols of the eastern Darling Downs, Queensland. I. Effects of surface conditions on soil movement within contour bay catchments. *Soil Research*, 24(2), pp.135-158.
- Ghahramani, A., Freebairn, D.M., Depika, S., Silburn, D.M., (Submitted, 2019) A pragmatic approach to estimating water quality leaving agricultural lands. *Environmental Modelling & Software*.
- Ghahramani, A. and Bowran, D., 2018. Transformative and systemic climate change adaptations in mixed crop-livestock farming systems. *Agricultural Systems*, 164, pp.236-251.
- Ghahramani, A., An-Vo, D.A., Rochecoste, J., Mushtaq, S., Robertson, S.D., Bennett, J. McL., Cockfield, G., 2018. Towards an economic decision support system for amelioration of soil constraints in the Northern Grains Region. USQ Centre for Sustainable Agricultural Systems Publication 1006897-18.
- Huth, N.I., Bristow, K.L. and Verburg, K., 2012. SWIM3: model use, calibration, and validation. *Transactions of the ASABE*, 55(4), pp.1303-1313.
- Holzworth, D.P., Huth, N.I., deVoil, P.G., Zurcher, E.J., Herrmann, N.I., McLean, G., Chenu, K., van Oosterom, E.J., Snow, V., Murphy, C. and Moore, A.D., 2014. APSIM– evolution towards a new generation of agricultural systems simulation. *Environmental Modelling & Software*, 62, pp.327-350.
- Knisel, W.G., 1980. CREAMS: a field scale model for Chemicals, Runoff, and Erosion from Agricultural Management Systems. United States Dept. of Agriculture. Conservation research report.
- Lerman, J., 2010. Programming Entity Framework: Building Data Centric Apps with the ADO. NET Entity Framework. " O'Reilly Media, Inc."

- Leonard, R.A., Knisel, W.G. and Davis, F.M., 1995. Modelling pesticide fate with GLEAMS. *European Journal of Agronomy*, 4(4), pp.485-490.
- Littleboy, M., Silburn, D.M., Freebairn, D.M., Woodruff, D.R., Hammer, G.L. and Leslie, J.K., 1992. Impact of soil erosion on production in cropping systems. I. Development and validation of a simulation model. *Soil Research*, 30(5), pp.757-774.
- Pechlivanidis, I.G., Jackson, B.M., McIntyre, N.R. and Wheeler, H.S., 2011. Catchment scale hydrological modelling: a review of model types, calibration approaches and uncertainty analysis methods in the context of recent developments in technology and applications. *Global NEST journal*, 13(3), pp.193-214.
- Pembleton, K., 2016. A review of the HowLeaky model: Applications, development, maintenance and future governance. A report for the Department of Environment and Science, Queensland Government.
- Queensland Government, 2019. HowLeaky Model V5 Technical Documentation: Version 1.06. Department of Natural Resources, Mines and Energy and the Department of Environment and Science.
- Silburn, D.M., 2003. Characterising Pesticide Runoff from Soil on Cotton Farms using a Rainfall Simulator. Unpublished Ph.D., Faculty of Agriculture, Food and Natural Resources, University of Sydney.
- Shaw, M., Silburn, M., 2016. Paddock to Reef Integrated Monitoring, Modelling and Reporting Program, Paddock Scale Modelling Technical Report. Department of Natural Resources and Mines, Queensland Government.
- Shaw, M., Silburn, M., Thornton, C., Robinson, B., McClymont, D., 2011. "Modelling pesticide runoff from paddocks in the Great Barrier Reef with HowLeaky." In MODSIM2011 19th International Congress on Modelling and Simulation, Sustaining our future: understanding and living with uncertainty, p. 2057. Modelling and Simulation Society of Australia and New Zealand, Perth.
- Šimůnek, J., Van Genuchten, M.T. and Šejna, M., 2006. The HYDRUS software package for simulating two-and three-dimensional movement of water, heat, and multiple solutes in variably-saturated media. Technical manual, version, 1, p.241.
- Sorooshian S., Gupta V.K., 1995. Model Calibration, Computer models of watershed hydrology, edited by Singh, V.P., Water Resources Publications, USA.
- The Queensland Government, 2016. Great Barrier Reef, report card 2016, reef water quality protection plan. Technical report.
- Robinson, J.B, Shaw, M., Viagak, V., Roberts, A., Thornton, C., Silburn, D.M., McClymont, D., 2009. An improved model for linking phosphorus loads in runoff to climate, soil and agricultural management 18th World IMACS/MODSIM Congress, Cairns, Australia.
- Williams, J.R., 1983. 'EPIC, The Erosion-Productivity Impact Calculator, Volume 1. Model Documentation.' Agricultural Research Service, United States Department of Agriculture.
- Wischmeier, W.H., Smith, D.D., 1978. Predicting rainfall-erosion losses - a guide to conservation planning.