The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume L-4/W1-2026
Free and Open Source Software for Geospatial (FOSS4G) 2026 – Academic Track “Bridging Geospatial Technology and Humanity”,
30 August–5 September 2026, Hiroshima, Japan

Development of an open-source analytical Digital Twin framework for environmental modelling and management

Luke Parkinson[1], Matthew Wilson[2, 3]

[1] Geospatial Research Institute, University of Canterbury - Christchurch, New Zealand - luke.parkinson@canterbury.ac.nz
[2] School of Earth and Environment, University of Canterbury - Christchurch, New Zealand - matthew.wilson@canterbury.ac.nz
[3] Geospatial Research Institute, University of Canterbury - Christchurch, New Zealand

Keywords: Digital Twin, Geospatial, TerriaJS, Modelling, Decision Making, Visualisation

Abstract

Environmental Digital Twins are dynamic virtual representations of physical environments that support real-time monitoring, scenario modelling, and community decision-making, that hold significant promise for addressing complex environmental challenges. Yet the development of each new system typically requires substantial duplicated engineering effort, and open-source frameworks specifically designed for environmental applications remain scarce. We present the **Environmental Digital Data Intelligence Engine (EDDIE)**, a free and open-source framework (AGPL-3.0) for building environmental Digital Twins, developed at the Geospatial Research Institute Toi Hangarau. EDDIE assembles proven free and open-source geospatial (FOSS4G) components including PostGIS, GeoServer, TerriaJS, and a Python processing stack into a modular, containerised architecture. A plugin-based module system allows domain-specific environmental models to be registered and orchestrated within a shared infrastructure, while OGC-standard service interfaces (WPS, WFS, WMS) ensure interoperability with external tools and other Digital Twin instances. EDDIE underpins three operational applications: the Flood Resilience Digital Twin (FReDT), which automates compound flood inundation modelling under climate change scenarios; the Ōtākaro Digital Twin, an environmental monitoring and modelling platform for the Ōtākaro/Avon River developed with hapū Ngāi Tūāhuriri and Christchurch City Council; and Te Awarua Kai Ora, a community hydrodynamic modelling and story-mapping platform for Te Awarua / Porirua Harbour. We describe the EDDIE architecture, present performance benchmarks for FReDT, and discuss EDDIE's potential as a reusable community framework for the open environmental Digital Twin ecosystem.

1. Introduction

1.1 Digital Twins

Digital Twins are software systems that provide dynamic virtual representations of physical systems, with automated data exchange and analytics being key attributes (Madni et al., 2019). The concept was first articulated in the context of aerospace simulation and has since evolved considerably: the defining characteristic of a mature Digital Twin is now understood to be bidirectional, persistent data exchange between a physical entity and its virtual counterpart, enabling not only monitoring and visualisation but also prediction, scenario testing, and informed decision-making (Fuller et al., 2020).

This definition distinguishes true Digital Twins from the related but more limited concepts of Digital Models (no automatic data flow) and Digital Shadows (one-way data flow from the physical to the virtual), a distinction increasingly emphasised in the literature as the term risks becoming diluted through broad usage (Kritzinger et al., 2018; Jones et al., 2020). Across all formulations, three capabilities are broadly agreed upon: (i) a persistent synchronisation mechanism that keeps the virtual representation current; (ii) analytical or simulation capacity that generates insight beyond raw data; and (iii) a pathway by which those insights can influence decisions about, or control of, the physical system (Fuller et al., 2020).

Digital Twins have proliferated rapidly, with applications spanning industrial manufacturing, smart cities (Li et al., 2021), infrastructure management, and, increasingly, the natural environment (Blair, 2021; Bauer et al., 2021). In environmental science the promise is especially significant: climate and Earth-system models are computationally intensive, their outputs are complex, and the gap between raw model data and actionable environmental insight is large. The EU's *Destination Earth* programme exemplifies the ambition at global scale, developing a planetary Digital Twin to democratise access to high-resolution weather and climate simulations for impact assessment and adaptation planning (Hoffmann et al., 2023).

1.2 Open-Source Environmental Digital Twin Frameworks

Despite the growth in individual Digital Twin implementations, the landscape of *reusable, open-source frameworks* specifically oriented to environmental applications remains sparse. Most published environmental Digital Twins are bespoke systems, constructed from scratch for a single domain or region, with limited capacity for re-use or extension (Gil et al., 2024). Surveys of open-source Digital Twin frameworks reveal that geospatial and environmental systems are underrepresented: existing open frameworks such as Eclipse Ditto or DTCC focus primarily on IoT connectivity or built-environment visualisation, rather than on the automated ingestion and processing of environmental data and the orchestration of physics-based models (Gil et al., 2024; Autiosalo et al., 2021).

A reusable framework for environmental Digital Twins would need to address several interrelated challenges. First, environmental models depend on heterogeneous, spatially referenced data from diverse open sources (LiDAR, gauge telemetry, satellite imagery, open government portals) that must be fetched, quality-controlled, and kept current automatically. Second, model runs are often computationally intensive but intermittent, demanding an architecture capable of scaling dynamically rather than maintaining dedicated compute capacity. Third, the outputs of environmental models are spatial; they need to be served and visualised in ways that are accessible to non-specialist stakeholders including community groups, planners, and policy makers. Finally, interoperability matters: as Autiosalo et al. (2021) describe, a growing "Digital Twin web" requires open standards so that individual twin instances can share data and compose into richer ecosystems.

The free and open-source geospatial community (FOSS4G) has developed many of the constituent building blocks — PostGIS for spatial data storage, GeoServer for standards-compliant spatial data services, TerriaJS for rich browser-based geospatial visualisation, and a mature Python scientific ecosystem — yet no open framework has previously assembled them into a coherent, extensible architecture aimed specifically at environmental Digital Twins.

This paper presents the **Environmental Digital Data Intelligence Engine (EDDIE)**, an open-source framework, released under the AGPL-3.0 licence and publicly available at github.com/GeospatialResearch/Digital-Twins, that addresses this gap.

1.3 Comparison with Existing Open-Source Digital Twin Frameworks

Several open-source Digital Twin frameworks already exist, and understanding where EDDIE differs from these is important context for its contribution. The most widely adopted general-purpose framework is Eclipse Ditto (Eclipse Foundation, 2017), which provides a robust IoT middleware layer for managing large networks of connected devices, representing their state as structured JSON objects accessible through REST and WebSocket APIs. Eclipse Ditto excels at device connectivity and state synchronisation, but it is not designed for the orchestration of physics-based environmental models, the ingestion of complex geospatial datasets, or the spatial visualisation of model outputs: capabilities that are central to EDDIE.

The Digital Twin City Centre (DTCC) platform and TerriaJS itself have been identified in the literature as the closest analogues in the geospatial Digital Twin space (Gil et al., 2024), both managing Digital Twins above a geospatial information layer. TerriaJS is, in fact, the front-end component of EDDIE, demonstrating the FOSS4G ethos of composing rather than duplicating existing tools. DTCC is oriented towards the built environment and urban digital twins, rather than the natural environment, and does not provide the modular environmental model orchestration pipeline that EDDIE offers. More recently, iTwin (Bentley Systems) provides sophisticated 3D infrastructure visualisation but is proprietary and focused on built-asset management rather than environmental modelling (Gil et al., 2024).

What distinguishes EDDIE from all of these frameworks is the combination of: (i) explicit support for the full pipeline from automated open-data ingestion through physics-based model execution to spatial results serving; (ii) a plugin architecture that allows new environmental model domains to be registered without modifying the core framework; and (iii) OGC-standards-based interoperability designed from the outset for the environmental and geospatial community. To the authors' knowledge, no existing open-source framework combines these three capabilities in a single, reusable system aimed at environmental Digital Twins.

1.4 EDDIE and its Applications

EDDIE was developed at the Geospatial Research Institute Toi Hangarau and assembles proven FOSS4G components into a modular, containerised architecture for building environmental Digital Twins. Its core design goals are: automated ingestion and caching of open spatial data; a plugin-based module system allowing domain-specific environmental models to be registered and orchestrated; standards-compliant service interfaces (OGC WPS, WFS, WMS) enabling interoperability with external tools and other Digital Twin instances; and an accessible web-based visualisation layer powered by TerriaJS.

EDDIE has been applied across multiple active research projects. The **Flood Resilience Digital Twin** (FReDT), the original application from which the EDDIE core was extracted, supports automated compound flood inundation modelling under user-specified climate change scenarios. The **Ōtākaro Digital Twin**, developed in collaboration with the hapū Ngāi Tūāhuriri and Christchurch City Council, provides an environmental monitoring and modelling platform for the Ōtākaro/Avon River corridor. Most recently, **Te Awarua Kai Ora** has been deployed for Te Awarua / Porirua Harbour, enabling community story-mapping and hydrodynamic flow modelling of the harbour and its catchment.

The remainder of this paper describes the EDDIE framework architecture (Section **??**), presents its deployment including FReDT as a detailed case study (Section 2), and discusses EDDIE's potential as a community framework for environmental Digital Twins (Section 3).

1.5 Design and Implementation

EDDIE provides a framework for creating Digital Twins through a Python library and a standardised set of containers. Implementations of EDDIE are comprised of multiple containers working together to form a web application. An implementation of EDDIE may be focused on a particular discipline, such as the Flood Resilience Digital Twin (FReDT) (Wilson et al., 2026) which can run flood inundation models according to users' needs. Alternatively, an implementation may be focused on a particular geographical area, as is the case with the Ōtākaro Digital Twin (Parkinson et al., 2024) and its focus on the Ōtākaro/Avon River in Christchurch, New Zealand. EDDIE is used to create Digital Twins with the capability to run models with customised scenario parameters, and to visualise the predictive outputs of these models alongside data from external web sources.

EDDIE applications consist of one or more domain-specific modules for running models. The application can either be run directly from one such module to make a domain-specific Digital Twin, or it can be run from the core EDDIE Python framework. Domain-specific modules are identified by their package name, the EDDIE framework searches for names starting with `"eddie_"` and their modelling pipelines can be automatically included in applications to form more complex Digital Twins.

The standard front-end for EDDIE applications is TerriaJS, with the back-end containers able to expose detailed dynamic catalogs. These catalogs can also be used by other independent

Digital Twins, enabling them to use all functionality available to create more powerful ecosystems of Digital Twins.

A typical EDDIE application is comprised of five core Open Container Initiative images, each with at least one service. However, the back-end image must provide at least two services, as shown in the following minimum list of services and images:

- Implementation-specific back-end image
  - Back-end API controller service
  - Celery Worker service
- TerriaMap image
- GeoServer image
- PostGIS image
- Redis image

EDDIE is designed around this containerised set of services to allow for more flexibility in the setup of EDDIE instances. This eases the setup new systems, and allows for more processing resources to be allocated as needed by starting additional Celery workers as needed with horizontal scaling or allocating more virtual CPUs or RAM as vertical scaling.

TerriaJS (TerriaJS Development Team, 2025), through the Terria Map image, is used as the primary front-end interface for users interacting with an EDDIE instance. TerriaJS is used to visualise complex geospatial data, pull data for visualisation from external sources, configure model inputs, and visualise model outputs. Data from the EDDIE PostGIS database can be fetched through GeoServer, and new model runs are requested through the back-end API controller service. TerriaJS supports 3D visualisations, opening the possibility for more immersive visualisations than is possible with 2D maps.

The back-end image is split into two main component services, the API controller, and the celery worker source code. The API controller uses the PyWPS (PyWPS Development Team, 2025) library to provide an implementation of the Open Geospatial Consortium (OGC) Web Processing Service standard (Open Geospatial Consortium, 2018) (WPS). This standardised interface allows other services to query the back-end for information on available processsing pipelines, and their required scenario input parameters. This allows requests to use existing tooling with standardised inputs, with results that can be used in further processing scripts or can be automatically displayed on the web. Once a request to run a model is received, the request is validated and encoded as a Python Celery task on the Redis message queue. This allows the processing to be handled by another service that may be running with more compute resources than the API controller requires.

The Celery worker service, whether a single worker or many, acts to complete the main processing pipeline of a modelling task. Celery allows for asynchronous tasks that may run in parallel depending on worker and workflow configuration. Tasks are taken from the Redis message queue and the scenario parameters are used to run models. Complex modelling pipelines may include automated retrieval of data from external APIs, so the EDDIE library can be used to facilitate downloading through standard interfaces such as OGC Web Feature Service (Open Geospatial Consortium, 2014) (WFS) for vector data and Web Map Service (Open Geospatial Consortium, 2006) (WMS) for raster tile data.

Complex pipelines may require multiple stages of fetching data, and then transforming them for use. EDDIE allows us to store the data in the PostGIS database while providing pre-checks for future processing steps that may allow further processing to skip a step if the correct data is already pre-cached. For example, a Light Detection and Ranging (LiDAR) point cloud to Digital Elevation Model (DEM) raster processing step may depend on a specific area of interest. If a subset of this area of interest is specified, EDDIE can fetch the corresponding subset of the pre-cached DEM without the costly LiDAR processing step. Once a model has completed, the results are saved to the PostGIS database, and the model results are configured as layers within GeoServer to be served. EDDIE provides functions to save and serve these results, and to set up a cache so future model executions may skip all processing steps if their scenario parameters sufficiently match those of a model scenario already run. Figure 1. shows the sequence of a model running in an EDDIE application, from querying the available models and their parameters, through to serving the resultant geospatial layers.

We use OGC standards for processing (WPS) and for serving data (WFS, WMS) to allow EDDIE Digital Twins to be interoperable with existing tooling. For example, users can export data from model results or link directly to GeoServer from desktop software such as QGIS or their own scripts to continue analysis beyond the capability of the included modelling.

2. Case studies and EDDIE deployment

2.1 Case study - Ōtākaro Digital Twin

EDDIE was used as the basis for the Ōtākaro Digital Twin, a prototype environmental platform for monitoring the health of the Ōtākaro/Avon River in Christchurch, New Zealand. This Digital Twin was created in collaboration with the local hāpu (sub-tribe), Ngāi Tūāhuriri and Christchurch City Council. Modelling available within the platform focuses on the MEDUSA 2.0 stormwater pollutant runoff model using user-inputted rainfall event parameters, and potential future modelling may include linking this to rainfall gauge telemetry. After configuring a rainfall event scenario, and running the MEDUSA 2.0 model, contributing pollutant sources are estimated and returned to the user. These are presented as geospatial polygon layers of road surfaces and roof surfaces, with their surface type predicted by a deep-learning model and summary of key pollutant loading variables such as "Total Suspended Solids" and "Disolved Copper" presented as outputs. Using TerriaJS to visualise, users can compare scenarios with a juxtaposition slider, and can style polygons according to the variables most important to their analysis. The Ōtākaro Digital Twin acts as a single interface that users can view and compare data from many data sources. Christchurch City Council's Open Data Portal items relating to river health are available within the Digital Twin, in addition to select layers from other open sources. The Ōtākaro Digital Twin prototype aims to collect and communicate knowledge about the Ōtākaro River and allows users to model potential futures. With more development this, Digital Twin will grow to collate more data and models to further support descision-making for the Ōtākaro.

2.2 Case Study - Flood Resilience Digital Twin

Flood risk assessment requires access to flood inundation models and spatial data on natural and built environment features that may be impacted by inundation. Creating these flood models is complex and labour-intensive. The Flood Resilience Digital Twin (FReDT) collates information on the natural and

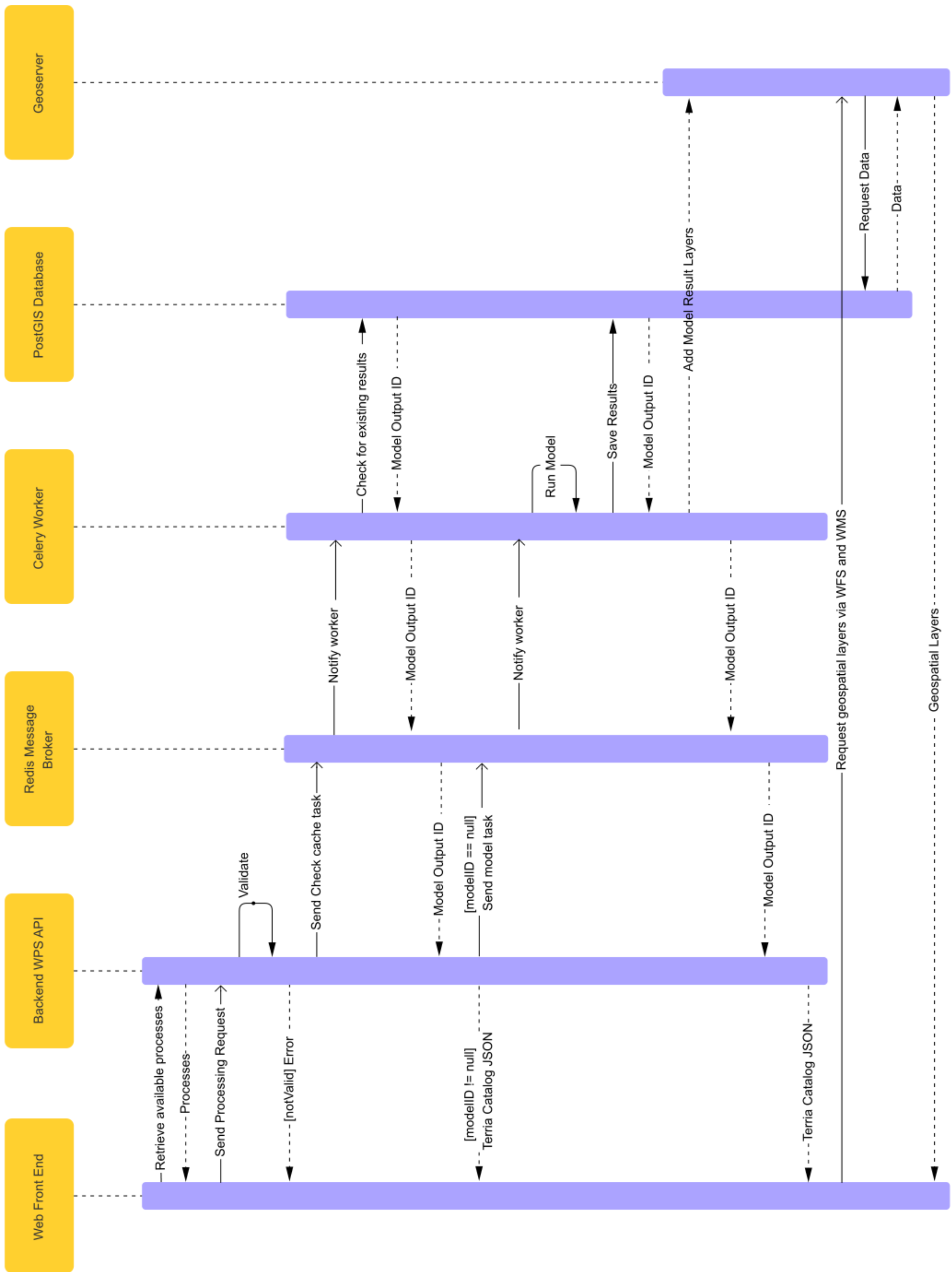

Figure 1. Sequence diagram showing events and data propagating through each EDDIE service for a modelling run request.

build environment and automatically runs flood model scenarios based on statistically-derived design storms and design hydrographs. This modelling requires large amounts of processing as discussed in Section 2.2.1. FReDT produces flood inundation layers for varying scenarios accounting for sea-level rise different rainfall intensities. Static flood maps do not offer the full details of a flood model output, and cannot easily be compared against infrastructure and natural environment layers for interactive assessment without geospatial skills. FReDT analyses the flood impact against buildings and presents the layers in an interactive way, as shown in Figure 2.

2.2.1 Benchmarks Figure 3 shows the significant performance gain that can be achieved due to appropriately caching process outputs. Initialising LiDAR metadata took over nine minutes and could be entirely skipped in all subsequent runs since the metadata was now up-to-date. Processing the LiDAR point cloud data into a hydrologically-conditioned DEM GeoFabric took over 14 minutes for the 6.7 km^2 area and approximately an hour for the 74.7 km^2 area. When running subsequent runs for the same area or an area within, the GeoFabric processing step can be instantly skipped. There are also performance gains in other stages of processing, such as preparing the dynamic boundary conditions data due to geospatial data from external sources being pre-stored and ready to process.

Had the scenario parameters used in the testing shown in Figure 3 been exactly the same as a previous scenario, then the entire processing pipeline would have been skipped, and the results would have been made served to the user immediately.

2.3 Deployment Requirements and Guidelines

Each domain-specific module for an EDDIE Digital Twin may require different external data sources or hardware capabilities. For example, the FReDT uses the BG-Flood (Bosserelle et al., 2021) flood model, which utilises GPGPU procressing to decrease the time required to run its numerical model. As such, it requires container access to the GPU via the CUDA Container Toolkit and an NVIDIA GPU. To deploy FReDT on a virtual private server or in the cloud, it is recommended to have access to an NVIDIA GPU for performance.

Another common requirement for deploying an EDDIE Digital Twin is a configuration file listing API keys for external data sources, such as the Ōtākaro Digital Twin and FReDT both requiring access to the Toitū Te Whenua Land Information New Zealand spatial data portal to retrieve data such as building outlines.

EDDIE has been developed and tested for Windows 10/11 and Ubuntu 22.04 operating systems. It is not recommended for use on MacOS due the lack of availability of image dependencies compiled for ARM64 CPU architectures and less testing completed for Mac as compared to Windows and Linux systems.

Once hardware, operating system, and API key requirements are met, Docker can be used to build the images. These can be deployed simply on a server using docker compose, which is perfectly suitable for demonstrations and development. For more robust but complex deployments, using Kubernetes may be advised. Kubernetes allows for orchestration of containers and their resources and further work towards this is discussed in Section 3.4.

3. Discussion and Summary

3.1 EDDIE as a Reusable Community Framework

The core contribution of EDDIE to the FOSS4G community is not any single Digital Twin application, but the framework itself: a structured, open-source foundation that lowers the cost of creating new environmental Digital Twins. Prior to EDDIE, a team wishing to build an environmental Digital Twin faced the full burden of assembling a software stack, designing data ingestion pipelines, implementing model orchestration, and building a web interface from scratch. EDDIE addresses this by providing these capabilities as a reusable library and a standardised container architecture, so that new projects need only implement the domain-specific modelling logic, as a named Python module, while inheriting the rest of the infrastructure.

EDDIE is constructed entirely from FOSS4G components: PostGIS, GeoServer, TerriaJS, Redis, Python, and Docker. It adopts OGC standards (WPS, WFS, WMS) at every service boundary, ensuring that data and processing results are consumable by any compliant client, including desktop GIS tools such as QGIS, scripting environments, or other Digital Twin instances. This interoperability is a precondition for the kind of federated “Digital Twin web” that researchers have identified as the trajectory of the field (Autiosalo et al., 2021): independent EDDIE instances can already expose their data catalogs to one another, forming nascent ecosystems of environmental twins.

The three operational applications demonstrated in this paper, FReDT, the Ōtākaro Digital Twin, and Te Awarua Kai Ora, span different environmental domains (flood resilience, urban river health, and harbour hydrodynamics) and different collaboration contexts (regional councils, indigenous hapū, community groups, and researchers). This breadth demonstrates that the EDDIE framework is genuinely domain-agnostic at its core, while the plugin module system allows deep domain specificity where it is needed. The FReDT benchmarking results show concretely how the caching architecture translates into practical performance gains: LiDAR pre-processing steps that would otherwise consume fourteen minutes to an hour or more are skipped entirely on subsequent runs, and identical scenario parameters yield instant results. These characteristics are important for interactive community engagement, where iterative scenario exploration in real time is essential to meaningful participation.

3.2 Licensing, Public Funding, and the FOSS4G Social Contract

EDDIE is released under the GNU Affero General Public Licence version 3 (AGPL-3.0), a choice that reflects both principled and practical considerations. The AGPL-3.0 is a strong copyleft licence: any derivative work, including software accessed over a network, must itself be released under the same terms. This was a deliberate decision motivated by the fact that EDDIE was developed using public funding. Software built with public resources should, in the view of the authors, remain publicly available; the AGPL-3.0 provides a legal mechanism to ensure that commercial actors who build upon EDDIE are required to contribute their modifications back to the commons, rather than enclosing publicly funded innovation in proprietary products.

This position aligns with a growing consensus in the research software community and among research funders that open,

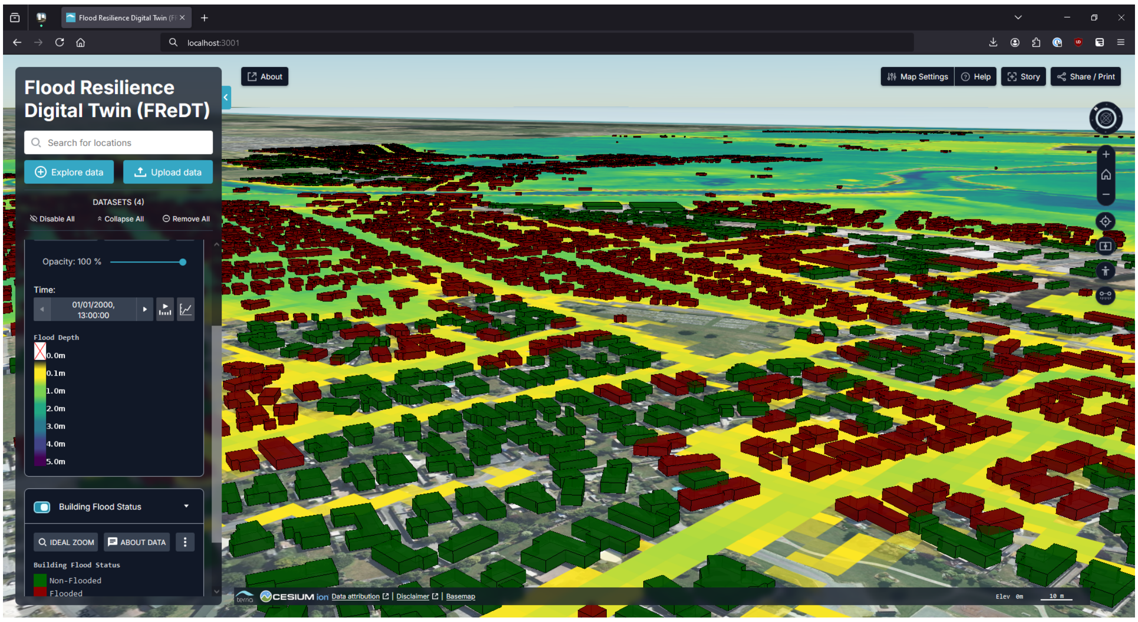

Figure 2. A screenshot of the Flood Resilience Digital Twin showing a flood inundation depth raster with building polygons overlaid and affected buildings coloured in red.

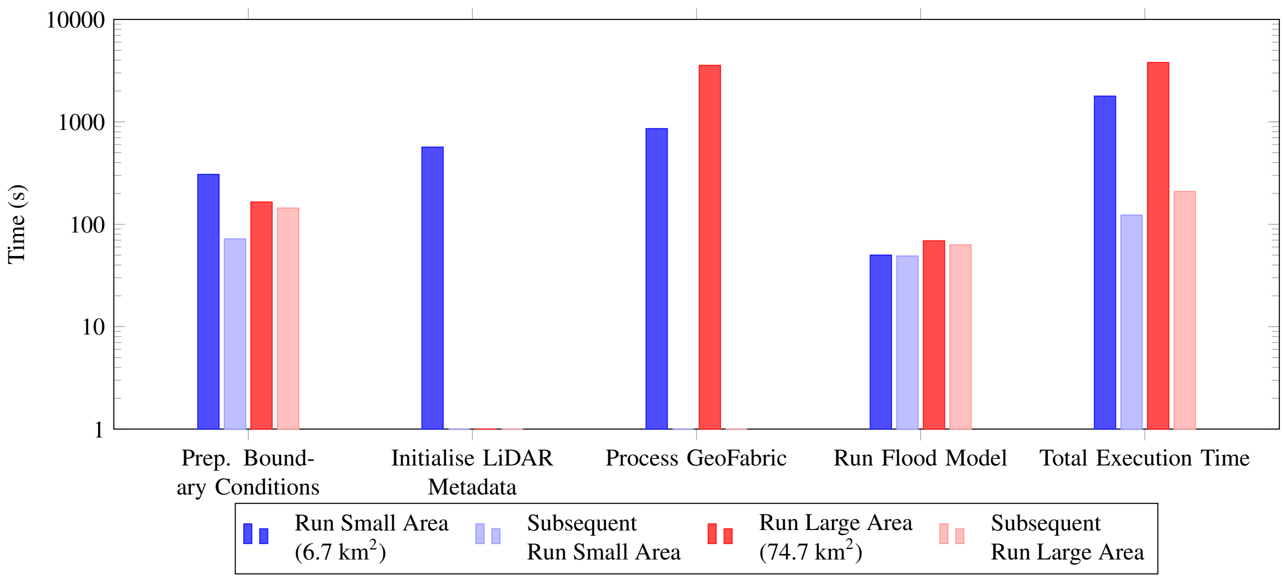

Figure 3. Execution time per Flood Resilience Digital Twin task stage across sequential run configurations. Larger areas take longer to execute but caching can eliminate the longest steps if the data is already present.

publicly funded software should adopt licences that prevent downstream re-enclosure (Stodden and Miguez, 2014; Stallman, 2006). For the FOSS4G community in particular, which has long navigated the tension between openness and commercial exploitation of open-source geospatial tools, the choice of a strong copyleft licence over a permissive one (such as MIT or Apache 2.0) signals a commitment to a reciprocal model of open development. Users of EDDIE are warmly encouraged to contribute improvements, new domain modules, and deployment configurations back to the project; the AGPL-3.0 ensures that this invitation is backed by legal obligation for those who build upon the framework commercially.

3.3 Community Co-design and Indigenous Collaboration

A distinguishing feature of the EDDIE applications described here is their grounding in genuine co-design with affected communities, and in particular with Māori partners who hold kaitiakitanga (guardianship, stewardship) over the environments being modelled. The Ōtākaro Digital Twin was developed in direct collaboration with Ngāi Tūāhuriri, the hapū with kaitiakitanga (guardianship) over the Ōtākaro/Avon River, and with Christchurch City Council. The FReDT project is actively embedding mātauranga Māori (Māori knowledge systems) within flood scenario design, so that nature-based solutions reflect Māori values and practices as well as technical hydrological considerations. This aligns with obligations under the United Nations Declaration on the Rights of Indigenous Peoples and with New Zealand’s te Tiriti o Waitangi commitments. The origins of this work lie in conversations with Te Rūnanga o Ngāi Tahu, the governing tribal council for Ngāi Tahu, the principal Māori iwi of New Zealand's South Island, whose leaders guided the framing of the digital twin concept as a tool for Māori environmental governance rather than merely a technical monitoring system.

This approach is grounded in the principles of Māori data sovereignty as articulated by Te Mana Raraunga, the Māori Data Sovereignty Network, which asserts that Māori data should be subject to Māori governance and should advance Māori self-determination and wellbeing (Te Mana Raraunga, 2016; Kukutai and Taylor, 2016). The six principles of Māori data sovereignty, Rangatiratanga (authority), Whakapapa (relationships), Whanaungatanga (obligations), Kotahitanga (collective benefit), Manaakitanga (reciprocity), and Kaitiakitanga (guardianship), provide a framework that extends and, in some respects, challenges mainstream open-data principles such as FAIR (Te Mana Raraunga, 2016). Whereas FAIR emphasises findability, accessibility, interoperability, and reusability as universal goods, the Māori data sovereignty framework foregrounds the question of *for whom* data is collected, governed, and made accessible, and requires that the communities whose environments are the subject of data retain meaningful authority over those data (Carroll et al., 2020).

The complementary CARE Principles for Indigenous Data Governance (Collective Benefit, Authority to Control, Responsibility, and Ethics), developed by the Global Indigenous Data Alliance, provide a parallel international framing that aligns with Te Mana Raraunga's principles and has been adopted by a growing number of research institutions (Carroll et al., 2020). EDDIE's co-design model operationalises several of these principles in practice: Rangatiratanga is reflected in partner communities holding decision-making authority over which data are surfaced and how results are presented; Kotahitanga is reflected in ensuring that Digital Twin outputs are accessible to community members, not only to technical specialists; and Kaitiakitanga is reflected in the primary purpose of each application, supporting the communities responsible for the long-term health of the environments being modelled.

We consider indigenous co-design not as an optional enrichment but as an epistemic and ethical requirement for environmental Digital Twins deployed in Aotearoa New Zealand. Digital systems that represent environmental conditions have direct implications for resource management, planning, and policy, domains in which Māori hold rights under te Tiriti o Waitangi. Designing Digital Twins without meaningful Māori partnership risks reproducing, in digital form, the exclusion of Māori from environmental governance that has characterised much of New Zealand's colonial history (Selby et al., 2010). We hope that the EDDIE framework, and the partnerships established in these projects, provide a model for how the FOSS4G community can engage with indigenous knowledge holders as genuine partners in the development of open environmental technology.

3.4 Limitations and Future Work

EDDIE is under active development and several limitations remain. The current deployment model is primarily based on `docker-compose`, which is well-suited to single-server deployments but limits the scalability available for computationally demanding scenarios. Ongoing work targets cloud-native deployments using Kubernetes and cloud-optimised data formats such as Zarr, which will enable dynamic horizontal scaling: large numbers of parallel scenarios can be run on cloud compute for the duration needed and then released, making large-scale community scenario testing economically viable. MacOS is not currently supported as a development platform, which may limit contributions from some developers.

Looking ahead, the EDDIE roadmap includes: integration of Apache Airflow for more robust workflow orchestration of multi-stage modelling pipelines; cloud-native data formats to reduce I/O overhead in large spatial domains; hybrid physics-AI modelling approaches to accelerate computationally expensive steps; and extension to new domains including an Antarctic Digital Twin. Virtual reality visualisation is also under exploration as a channel for community engagement with complex three-dimensional environmental model outputs. As EDDIE matures and its deployment templates lower the barrier to adoption, we hope the framework will attract contributors from the broader FOSS4G and environmental modelling communities, strengthening the shared infrastructure available to all.

3.5 Summary

Environmental Digital Twins have the potential to transform access to modelling and data for communities, planners, and researchers grappling with pressing environmental challenges. Yet the absence of open, reusable frameworks has meant that each new implementation requires substantial duplicated engineering effort, limiting both the pace of adoption and the interoperability of the systems produced.

EDDIE addresses this gap directly: a free, open-source (AGPL-3.0) framework that assembles proven FOSS4G components into a coherent architecture for environmental Digital Twins, built on open standards and designed for extensibility. Its three operational applications of flood resilience, urban river health, and harbour hydrodynamics, demonstrate the framework's capacity to serve diverse domains and to engage communities as genuine participants in environmental insight. We invite the FOSS4G community to adopt, extend, and contribute to EDDIE, and we hope it provides a foundation on which a richer, federated ecosystem of open environmental Digital Twins can be built.

4. Conclusion

Environmental Digital Twins offer a powerful mechanism for bridging the gap between complex environmental models and the communities, planners, and researchers who need to act on their insights. The central obstacle to their wider adoption is not conceptual but practical: the absence of reusable, open-source infrastructure means that each new implementation begins from scratch, duplicating engineering effort that could instead be directed at domain-specific modelling and community engagement. EDDIE addresses this directly.

By assembling established FOSS4G components into a coherent, extensible architecture built on OGC open standards, EDDIE provides a foundation on which new environmental Digital Twins can be constructed with substantially reduced overhead. The plugin module system means that domain expertise, such as in flood hydrology, stormwater quality, harbour hydrodynamics, or any other environmental domain, can be encapsulated in a self-contained Python package and immediately integrated into the full EDDIE infrastructure, including data ingestion, model orchestration, result serving, and web-based visualisation.

The three operational applications presented here (FReDT, the Ōtākaro Digital Twin, and Te Awarua Kai Ora) span distinct environmental domains and community contexts, demonstrating that this architecture is genuinely generalisable. The FReDT benchmarks demonstrate that intelligent caching of geospatial processing steps delivers the response times necessary for real-time community scenario exploration, turning what would otherwise be hours of computation into near-instant results for subsequent runs.

Co-design with Māori communities, grounded in the principles of Māori data sovereignty and the CARE framework, has been central to the development of these applications and has shaped both the technical architecture and the governance of data within the platform. We regard this not as a feature specific to the Aotearoa New Zealand context but as a model for how open environmental technology can be developed in genuine partnership with the communities whose environments are at stake.

EDDIE is publicly available at github.com/GeospatialResearch/Digital-Twins under the AGPL-3.0 licence. We invite the FOSS4G and environmental modelling communities to adopt, extend, and contribute to the framework, and we look forward to a growing ecosystem of interoperable open environmental Digital Twins built upon shared, community-maintained infrastructure.

References

Autiosalo, J., Siegel, J., Tammi, K., 2021. Twinbase: Open-Source Server Software for the Digital Twin Web. *IEEE Access*, 9, 140779–140798.

Bauer, P., Stevens, B., Hazeleger, W., 2021. A digital twin of Earth for the green transition. *Nature Climate Change*, 11(2), 80–83.

Blair, G. S., 2021. Digital twins of the natural environment. *Patterns*, 2(10), 100359.

Bosserelle, C. D., Lane, E. M., Harang, A., 2021. BG-Flood: A GPU adaptive, open-source, general inundation hazard model. *Proceedings of the Australasian Coasts & Ports 2021 Conference*, Christchurch, New Zealand.

Carroll, S. R., Garba, I., Figueroa-Rodríguez, O. L., Holbrook, J., Lovett, R., Materechera, S., Parsons, M., Raseroka, K., Rodriguez-Lonebear, D., Rowe, R., Sara, R., Walker, J. D., Anderson, J., Hudson, M., 2020. The CARE Principles for Indigenous Data Governance. *Data Science Journal*, 19, 43.

Eclipse Foundation, 2017. Eclipse Ditto: Open source framework for IoT digital twins. Eclipse Foundation. eclipse.dev/ditto (15 June 2026).

Fuller, A., Fan, Z., Day, C., Barlow, C., 2020. Digital Twin: Enabling Technologies, Challenges and Open Research. *IEEE Access*, 8, 108952–108971.

Gil, D., Johnsen, S. et al., 2024. Survey on open-source digital twin frameworks – a case study approach. *Software: Practice and Experience*.

Hoffmann, J., Bauer, P., Sandu, I., Wedi, N., Geenen, T., Thiemert, D., 2023. Destination Earth – A digital twin in support of climate services. *Climate Services*, 30, 100394.

Jones, D., Snider, C., Nassehi, A., Yon, J., Hicks, B., 2020. Characterising the Digital Twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology*, 29, 36–52.

Kritzinger, W., Karner, M., Traar, G., Henjes, J., Sihn, W., 2018. Digital Twin in Manufacturing: A Categorical Literature Review and Classification. *IFAC-PapersOnLine*, 51(11), 1016–1022.

Kukutai, T., Taylor, J. (eds), 2016. *Indigenous Data Sovereignty: Toward an Agenda*. ANU Press, Canberra.

Li, D., Yu, W., Shao, Z., 2021. Smart city based on digital twins. *Computational Urban Science*, 1(1), 4.

Madni, A. M., Madni, C. C., Lucero, S. D., 2019. Leveraging Digital Twin Technology in Model-Based Systems Engineering. *Systems*, 7(1), 7.

Open Geospatial Consortium, 2006. OpenGIS® Web Map Server Implementation Specification. Technical Report OGC® 06-042, Open Geospatial Consortium, Arlington, VA., United States.

Open Geospatial Consortium, 2014. OGC® Web Feature Service 2.0 Interface Standard. Technical Report 09-025r2, Open Geospatial Consortium, Arlington, VA., United States.

Open Geospatial Consortium, 2018. OGC® WPS 2.0.2 Interface Standard. Technical Report 14-065r2, Open Geospatial Consortium, Arlington, VA., United States.

Parkinson, L., Li, C., Nguyen, M., Charters, F., Thompson, S., McIntyre, A., Gray, K., 2024. Ōtākaro Digital Twin Software, Version 1.0. GitHub github.com/GeospatialResearch/Otakaro-Digital-Twin (29 November 2024).

PyWPS Development Team, 2025. PyWPS Software, Version 4.7.0. PyWPS. pywps.org (15 December 2025).

Selby, R., Moore, P., Mulholland, M. (eds), 2010. *Māori and the Environment: Kaitiaki*. HUIA Publishers, Wellington, New Zealand.

Stallman, R. M., 2006. *Free Software, Free Society: Selected Essays of Richard M. Stallman*. Free Software Foundation, Boston, MA.

Stodden, V., Miguez, S., 2014. Best Practices for Computational Science: Software Infrastructure and Environments for Reproducible and Extensible Research. *Journal of Open Research Software*, 2(1), e21.

Te Mana Raraunga, 2016. Te Mana Raraunga charter. Māori Data Sovereignty Network. temanararaunga.maori.nz (15 June 2026).

TerriaJS Development Team, 2025. TerriaJS Software, Version 8.9.3. Terria Pty Ltd. terria.com/open-source (24 April 2025).

Wilson, M., Parkinson, L., Li, C., Khosla, P., Nguyen, M., Pearson, R., 2026. Flood Resilience Digital Twin (FReDT) Software, Version 3.0.0. GitHub github.com/GeospatialResearch/eddie_floodresilience (8 April 2026).