

# Software Design and FAIR Principles: Bridging the Gap (Up to 15 words)

Eirian O Perkins  
New Zealand eScience Infrastructure  
eirian.perkins@nesi.org.nz

## ABSTRACT / INTRODUCTION (Up to 200 words)

Findable, accessible, interoperable, and reusable (FAIR) software is achievable when the design phase is driven by software engineering principles, namely software design patterns. Incorporating these principles is as significant as understanding researcher needs when implementing tools to support academic endeavours. The Aotearoa Genomic Data Repository (AGDR), built on Gen3 technology, allows researchers to browse and upload taonga species data, and has a reasonably comprehensive metadata schema in order to ensure the data are FAIR and CARE - have collective benefit, Authority to control, Responsibility and Ethics. One of the implementation challenges was to validate metadata for ingest and provide sensible feedback to the user. The validator was built in Python and takes advantage of structural and behavioural design patterns, and is loosely coupled with AGDR-specific implementations so that the tool can be shared back to the wider Gen3 community. While designing software to support FAIR principles seems obvious, emphasizing user experience over software design or vice versa may result in outputs that are neither interoperable nor reusable.

## ABOUT THE AUTHOR(S)

- Eirian is a devops and integration programmer at the New Zealand eScience Infrastructure, as well as a PhD candidate at the University of Auckland in the School of Biological Sciences and has over 10 years of combined industry and public sector experience.