



**Legislative Assembly** for the  
**Australian Capital Territory**

Standing Committee on Environment,  
Planning, Transport and City Services

# Submission Cover Sheet

## Inquiry into the procurement and delivery of MyWay+

Submission number: 056.2

Submitter: Patrick Reid - Supplementary

Date authorised for publication: 9 May 2025

**From:** [Patrick Reid](#)  
**To:** [LA Committee - Environment](#)  
**Subject:** Supplementary Submission - EOL software in Customer Portal - Inquiry into the procurement and delivery of MyWay+  
**Date:** Monday, 5 May 2025 3:40:11 PM

**Caution:** This email originated from outside of the ACT Government. Do not click links or open attachments unless you recognise the sender and know the content is safe.

Good afternoon,

I was watching the inquiry from Thursday and noted that NEC made factually incorrect claims about end of life software components in MyWay+. I would like to provide a supplementary submission outlining why I believe this is false.

Mr Messenger claimed that, because the system was based off a SaaS service, there was no end of life components. I believe that this is intended to mean that no NEC software components are end of life. However, that ignores third-party components (dependencies) that are close to or are already at end of life. Consider this table which contains a subset of the dependencies used as part of the customer portal front-end<sup>[^1]</sup>.

Library	Library owner	Used version (date)	Latest version	Used is EOL	Has minor <sup>[^2]</sup> non-EOL	Has major <sup>[^2]</sup> non-EOL	Has post-EOL commercial security support	NEC pays for commercial support
<a href="#">linq.js</a>	<a href="https://neue.cc/">https://neue.cc/</a>	2.2.0.0 (2010)	3.0.4-Beta5 (2013, abandoned) or 4.0.3 (2024, fork by different author)	Yes - Repo is archived, no stated maintenance.	No	If you use the fork	No	-
<a href="#">jQuery</a>	OpenJS foundation formerly jQuery foundation	1.11.3 (2015)	3.7.1 (2023)	Yes	No	Yes	Yes	No
<a href="#">jQuery UI</a>	OpenJS foundation formerly jQuery foundation	1.12.1 (2016)	1.14.1 (2024)	Yes	Maintenance only	Maintenance only	No	-
<a href="#">littlepay</a>	<a href="https://littlepay.com/">https://littlepay.com/</a> ?	?	?	?	?	?	?	?
<a href="#">bootstrap</a>	Bootstrap core contributors, formerly Twitter	3.3.7 (2016)	5.3.5 (2025)	Yes	No	Yes	Yes	No
<a href="#">moment.js</a>	Momentjs contributors	2.13.0 (2016)	2.30.1 (2023)	?	Maintenance only	Maintenance only	?	?
<a href="#">bootstrap-datetimepicker</a>	Eonasdan	4.17.37	6.10.3 (2025)	Yes	No	Maintenance only	Sort of - issues must be paid for before fixed	No
<a href="#">sweetalert</a>	Tristan (t4t5)	?	2.1.0 (2017)	?	No	No	No - use sweetalert2 (different project)	-
<a href="#">bootstrap-timepicker</a>	jdewit	0.5.2 (2016)	0.5.2 (2016)	Yes	No	No	No	-
<a href="#">jspdf</a> and plugins	Various contributors	?(2016)	3.0.1 (2025)	Yes	?	Yes	?	?
<a href="#">pdf.js</a> as part of jspdf	<a href="https://mozilla.org">https://mozilla.org</a>	?(2011)	5.2.133 (2025)	Yes	?	Yes	No	-
<a href="#">jspdf autotable</a>	Simon Bengtsson (simonbengtsson)	2.3.2 (2014)	5.0 (2025)	Yes	No	Yes	No	-
<a href="#">data tables</a>	SpryMedia Ltd	1.10.15 (2017)	2.3.0 (2025)	Yes	No	Yes	Yes	No
<a href="#">bootstrap-select</a>	SnapAppointments	?	1.13.18 (2020)	?	?	<a href="#">superposition of EOL and not EOL</a>	No	-

This table is a bit of a mess, let me try to explain it:

- **Library** contains the name of the library and (if I can find it) a link to its source code
- **Library owner** as close as I can find to the person or organisation who owns the library. Using any of these libraries implies that you trust the author to not embedded malware into the libraries (very uncommon, but [happens](#))

occasionally)

- **Used version** is the version and year of the library used
  - Applications generally use the version from the year they were developed. This implies that the customer portal for MyWay+ (or the templates its built of) was initially created in 2016/2017
- **Latest version** is the latest release of the project I can find
- **Used is EOL** is if I consider the version to be EOL. I am judging this based on:
  - Has the project made a statement about the version's status (e.g. jQuery v1)
  - If there is a new major version, I consider the last major version(s) as EOL (e.g. bootstrap-datepicker-time)
  - If the author has abandoned the project, I consider all versions EOL (e.g. sweetalert)
- **Has minor non-EOL** or **Has major non-EOL** tells you if the library has a major or minor<sup>[^2]</sup> version bump that is not EOL.
  - If a library has a minor non-EOL, upgrading to a non-EOL should be as little as half an hour
  - Major version bumps may take significantly longer depending on the size of the project
- **Has post-EOL commercial security support** is yes if there is a commercial package that will provide you with security fixes.
- **NEC pays for commercial support** is yes if I think NEC is paying for commercial support. I would expect them to be on a higher minor version than publicly available

Some of these details are unknown. I have filled these in with "?". I could likely find an answer given enough time and energy (and NEC developers should know the answer to these questions).

This is not a complete audit of the libraries used by NEC. I do not have access to any of the back-end code or infrastructure configuration. The front-end also includes a bundle that does not have the required license information<sup>[^3]</sup>. If you know what you are looking for, you can identify major libraries like Angular. However, it is difficult to identify smaller libraries and the specific version of anything. This leads me to believe that my list of software is incomplete.

Regards,  
Patrick Reid

[^1]: Front-end here means the part of the UI the user can see. Different from the back-end, where data is processed and (possibly) stored.

[^2]: Most software libraries generally follow the [semantic versioning pattern](#). This prescribes the following rough version structure of Major.Minor.Patch. You can generally upgrade patch and minor changes without any code changes.

[^3]: Open source (freely available) code almost always has a license attached. This tells you what you can do with the code and how you should provide value back to the code author. This is often as simple as citing the author, although can be more extreme for copy-left licenses. We know NEC uses Angular, because we can see code that is part of Angular in their bundle. However, NEC does not credit Angular, even though [the license requires it](#). This is scummy and possibly illegal (on the same level as piracy), but minor.