What I Learned

This week I learned that open-source licenses create different obligations for users and developers, and those obligations connect to larger questions of intellectual property and ethics. In *Ethics in Technology*, intellectual property is described as intangible, transferable, and eventually part of the public domain. U.S. copyright (life + 70 years) and patents (20 years) are long compared to the fast pace of software development, raising fairness concerns (Weber, 2025, Ch. 10). Weber explains that IP law is justified to balance incentives for creators with benefits to society, but in the digital economy it can also create power imbalances.

Looking at license choices, the **MIT license** is the most permissive: it only requires attribution and inclusion of the license text. The **GPL** is a strong copyleft license, meaning any distributed derivative must also be under GPL. This protects user freedom but can limit adoption in proprietary projects (Free Software Foundation, n.d.). The **Apache 2.0 license** is permissive like MIT but adds explicit patent provisions, which help reduce legal risk for companies (Apache Software Foundation, 2004). The *Choose a License* resource gave a clear, side-by-side comparison (GitHub, n.d.), and OSI confirmed that all three are approved (Open-Source Initiative, n.d.). In Chapter 5, Weber discussed white-hat roles and coordinated disclosure, which shows me that license choices and disclosure practice both reflect professional responsibility: balancing freedom, fairness, and harm reduction (Weber, 2025, Ch. 5).

How I'll Apply It

For a small internal automation tool, I would choose the **MIT license** because:

- **Distribution:** If we later share the tool, MIT encourages adoption with minimal restrictions.
- **Compatibility:** MIT avoids conflicts with proprietary or copyleft software.
- **Obligations:** Attribution is the only requirement, which keeps compliance simple.

Muddiest Point

I am still unsure about the scope of GPL's copyleft. If a proprietary program links to a GPL library, does that mean the whole program must be GPL? The FSF says yes, but many developers treat dynamic linking as an exception (Free Software Foundation, n.d.). I am also unclear about dual licensing: if a project is GPL but also sold under a commercial license, does the commercial license fully remove GPL obligations for that user? I also wonder how disclosure rules, such as DMCA §512 safe-harbor protections, apply when researchers report vulnerabilities in detail (U.S. Copyright Office, n.d.).

Portfolio Note

• **License Reflection –** Shows I can compare MIT, GPL, and Apache in terms of obligations, ethics, and workplace application.

• **Responsible Disclosure Outline –** Demonstrates I understand CISA's VDP norms and how to report vulnerabilities safely.

AI Use Note

I used ChatGPT for brainstorming ideas and to clarify definitions of key concepts such as attribution, copyleft, and patent terms.

References

Apache Software Foundation. (2004). *Apache license, version 2.0.* https://www.apache.org/licenses/LICENSE-2.0

Free Software Foundation. (n.d.). Frequently asked questions about the GNU licenses. https://www.gnu.org/licenses/gpl-faq.html

GitHub. (n.d.). *Choose an open-source license.* https://choosealicense.com/

Open-Source Initiative. (n.d.). Approved licenses. https://opensource.org/licenses

U.S. Copyright Office. (n.d.). Section 512 of Title 17: Safe harbor provisions for online service providers. https://www.copyright.gov/512/

Weber, E. (2025). *Ethics in technology* (OER ed.).