Licensing Model 1: Dual Licensing (GPL v3 + Commercial License)

Description:
Hold companies hostage with a strong copyleft license forcing them to buy our commercial license to not infect their proprietary codebase.
Pros:

- Well-established and widely understood licenses (GPL v3 and commercial licensing)
- Generates revenue through commercial licensing for companies using the software in a closed-source fashion
- Allows us to use and modify GPL-based libraries

Cons:

- Resistance from companies averse to the strict terms of the GPL, limiting adoption
- Gives up control over commercial derivatives
- The idea of "holding companies hostage" with GPL doesn't sit well with me

Examples:

- MySQL (GPL + Commercial licenses)

Licensing Model 2: Opencore (MIT/Apache + Commercial offering)

Description:
Release the software under the permissive MIT/Apache License, but add additional closed-source services for which companies pay.
Pros:

- Well-established and widely understood licenses (MIT/Apache and commercial licensing)

Cons:

- I personally believe all features should be accessible, and pricing based on usage
- Harder to build a commercial product as you can't charge for core platform usage + allows for competitive commercial derivative works
- Need to maintain multiple codebases

Examples:

- Gitlab (MIT + Commercial offering)

Licensing Model 3: Source-Available License (BSL 1.1)

Description:
Adopt a source-available license like the MariaDB or Hashicorp. BSL 1.1, allows for internal modifications but prohibits production use and redistribution of modified versions without a commercial license. Also, this license becomes Apache/MIT after 4 years.
Pros:

- Maintains control over production usage and redistribution of derivative commercial products by requiring a commercial license
- Simpler to manage a single codebase compared to dual licensing models

Cons:

- There is no standard BSL licenses, meaning adoption may be stifled by OSS puritans or legal departments
- Has gotten a bad reputation over the past years as companies have shifted from OSS to this license
- Likely to receive fewer external contributions

Examples:

- HashiCorp (Terraform is licensed under the BSL 1.1)
- MariaDB MaxScale (BSL 1.1)

Licensing Model 1: Dual Licensing (GPL v3 + Commercial License)

Description:
Hold companies hostage with a strong copyleft license forcing them to buy our commercial license to not infect their proprietary codebase.

Pros:

- Well-established and widely understood licenses (GPL v3 and commercial licensing).
- Generates revenue through commercial licensing for companies using the software in a closed source fashion.
- Allows us to use and modify GPL based libraries.

Cons:

- Resistance from companies averse to the strict terms of the GPL, limiting adoption.
- Gives up control over commercial derivatives.
- I don't like the idea of holding companies hostage with GPL

Examples:

- Red Hat (GPL for Red Hat Enterprise Linux, commercial licenses for support and services)
- MySQL (GPL + Commercial licenses)

Licensing Model 2: Opencore (MIT/Apache + Commercial offering)

Description:
Release the software under the permissive MIT/Apache License, but add additional closesourced services for which companies pay for.
Pros:

- Well-established and widely understood licenses (MIT/Apache and commercial licensing)
-

Cons:

- I think every company should have access to all features, and they should be priced based on usage.
- Harder to build a commercial product as you can't charge for core platform usage + allows for competitive commercial derivative works.
- need to maintain multiple codebases.

Examples of Companies:

- Gitlab (MIT + Commercial offering

Licensing Model 3: Source-Available License (BSL 1.1)

Description:
Adopt a source-available license like the MariaDB or Hashicorp. BSL1.1, allows for internal modifications but prohibits production use and redistribution of modified versions without a commercial license. Also, this license becomes Apache/MIT after 4 years.
Pros:

- Provides transparency by making the source code available for reference and review.
- Maintains control over production usage and redistribution of derivative commercial products by requiring a commercial license.
- Simpler to manage a single codebase compared to dual licensing models.

Cons:

- There is no standard for BSL licenses, meaning adoption may be stifled by OSS puritans or legal departments.
- Has gotten bad wrap over the past years as companies have shifted from OSS to this license
- We will get fewer external contributions

Examples of Companies:

- HashiCorp (Terraform is licensed under the BSL 1.1)
- MariaDB max scale (BSL 1.1)

Licensing Model 1: Dual Licensing (GPL v3 + Commercial License)

Description:
Release the core codebase under the GNU General Public License (GPL) v3, a strong copyleft license. Additionally, offer a separate commercial license for organizations that want to create proprietary derivative works or commercial redistributions.
Pros:

- Allows non-commercial use, modification, and distribution under the GPL v3 terms, promoting open source collaboration.
- Retains control over commercial derivatives by requiring a commercial license for such use cases.
- Well-established and widely understood licenses (GPL v3 and commercial licensing).
- Generates revenue through commercial licensing for companies using the software commercially.

Cons:

- Potential fragmentation due to the existence of multiple GPL-based derivatives alongside the commercial versions.
- Complexity in maintaining separate open source and commercial codebases or branches.
- Some resistance from companies averse to the strict terms of the GPL v3, potentially limiting adoption.

Examples of Companies:

- Red Hat (GPL for Red Hat Enterprise Linux, commercial licenses for support and services)
- MySQL (GPL + Commercial licenses)
- Qt (LGPL + Commercial licenses)

Licensing Model 2: Opencore MIT/Apache + Commons Clause

Description:
Release the software under the permissive MIT License, but append the Commons Clause, which restricts commercial use and redistribution of modified versions without a separate commercial license.
Pros:

- Allows open source use, modification, and non-commercial redistribution under the permissive MIT terms.
- Retains control over commercial use and redistribution of modified versions through the Commons Clause.
- Simpler to maintain a single codebase compared to dual licensing models.
- Potentially more appealing to companies than copyleft licenses like GPL.

Cons:

- The Commons Clause is a relatively new addition, and its legal implications are still being explored, which may raise concerns.
- Perceived by some as restricting open source freedoms by limiting commercial use and redistribution.
- Might face resistance from companies or individuals who prefer unrestricted open source licenses.

Examples of Companies:

- MongoDB (Server Side Public License, which is similar to GPL + Commons Clause)
- Elastic (Apache 2.0 + Commons Clause for some products like Elasticsearch and Kibana)
- Redis (BSD + Commons Clause)

Licensing Model 3: Source-Available License (e.g., BSL 1.1)

Description:
Adopt a source-available license like the Hashicorp Source Available License (BSL) 1.1, which allows internal modifications but prohibits redistribution of modified versions without a commercial license.

Pros:

- Provides transparency by making the source code available for reference and review.
- Allows organizations to modify the software for their internal use without needing a commercial license.
- Maintains control over redistribution and creation of derivative commercial products by requiring a commercial license.
- Simpler to manage a single codebase compared to dual licensing models.

Cons:

- Perceived by some as restricting open source freedoms by limiting redistribution and derivative works.
- May face resistance from individuals or organizations that prefer traditional open source licenses.
- Potential for misalignment with the open source philosophy, as source-available licenses are considered more restrictive than open source licenses.

Examples of Companies:

- HashiCorp (Terraform is licensed under the BSL 1.1)
- Elastic (Elasticsearch and Kibana are licensed under the Elastic License, which is similar to BSL)

It's important to note that the choice of licensing model depends on your specific goals, business model, and the level of control you want to maintain over your software and its ecosystem. It's advisable to consult with legal counsel to ensure compliance and alignment with your requirements. Additionally, consider the potential impact on community adoption, open source philosophy, and commercialization opportunities when evaluating these licensing models.