Opening:

Privacy policy: inform visitors as to how information will be collected

Terms of service or Acceptable use policy: how users should be using the websites or software, not necessarily need to be accepted or decline because by using the website you’re agreeing to it. Basically a license but for a website.

Service level agreement (SLA): Outlines the service that is going to be provided

What is a software license also called EULA which stands for end user license agreement? Accepting gives the user the right to use your software on the condition they comply with all the terms.

Things that need to be considered when choosing:

* Commercial use: Whether you can use the software for commercial reasons
  + This clause is usually very simple and straightforward, it simply says whether or not you can use the software for commercial purposes.
    - License that allows: Apache license – Used by Apache, Android and Swift
    - License that doesn’t allow: Non-Profit Open Software License (no popular examples of use) Creative commons license – for websites is similar. Used by flickr, Wikipedia, bandcamp and youtube. Doesn’t really work for software, has caused a lot of issues on the website.
* Modify code: Ability to modify software
  + This clause states whether you can modify the software. Doesn’t usually place restrictions on modifying the software but on the distribution of modified code
    - License that allows: GNU General Public License v3 – Very common license with varying versions that allow/disallow different things, used by things such as the Linux Kernel, MySQL, Blender and VLC. Used to be a much more common license – has to do with redistribution of code.
    - License that doesn’t allow: Creative Commons No Derivatives – One of the creative commons forks. So you can use the software for commercial use but can’t modify it. It doesn’t really
* Redistribution of derivative works: Can distribute original or modified code (must also be ok to modify code)
  + This is just about whether or not code and derivated code can be redistributed, this is the most complicated clause because it can have a lot of stipulations which causes it to be infringed a lot.
    - There are licenses that don’t allow re distribution like the creation commons no distribution license.
    - It’s pretty standard for most licenses to require that the current license extends to all redistributions. It’s also pretty standard for most licenses to require a changelog that is included. I haven’t seen a license that doesn’t require all relevant copyright information to be included or one that doesn’t require any relevant licenses to be included.
    - A stricter license like the newest version of the GNU general public license has a stipulation that source code must be available for no more than the cost of the binaries. It also stipulated that install instructions must be included if part of a device. Now this includes how the software can be installed, reinstalled and modified.
* Warranty and liability: Ability to place a warrant on licensed software. Liability is whether or not consumers or contributors (in the case of open source software) can be held accountable for damages.
  + Warranty just means there’s protection in case of damages. Licenses that allow for warranties don’t include warranties because any warranty in a license would be too general they just allow developers to make their own.
  + Most licenses that have a warranty clause allow for sublicensing which means that more clauses or even whole licenses can be added to derivative works.
  + There are usually some stipulations that come with this. For example liability a lot of licenses don’t allow for a liability clause. Which makes a lot of sense because the liability would be on the developer to pay for a consumer misusing their product. There are obscure licenses that allow this but I don’t see why anyone would use them.
* Private use: Ability to use and modify software without distributing